

**Direction des bibliothèques**

**AVIS**

Ce document a été numérisé par la Division de la gestion des documents et des archives de l'Université de Montréal.

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

**NOTICE**

This document was digitized by the Records Management & Archives Division of Université de Montréal.

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal

Segmentation non-supervisée d'images couleur par  
sur-segmentation Markovienne en régions et  
procédure de regroupement de régions par graphes  
pondérés

par

Rachid Hedjam

Département d'informatique et de recherche opérationnelle  
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures et postdoctorales  
en vue de l'obtention du grade de  
Maître ès sciences (M.Sc.) en Informatique

Octobre, 2008

©Rachid Hedjam, 2008



**Université de Montréal**  
Faculté des études supérieures et postdoctorales

Ce mémoire intitulé  
**Segmentation non-supervisée d'images couleur par  
sur-segmentation Markovienne en régions et procédure de  
regroupement de régions par graphes pondérés**

présenté par :  
**Rachid Hedjam**

a été évalué par un jury composé des personnes suivantes:

**Philippe Langlais**

---

président-rapporteur

**Max Mignotte**

---

directeur de recherche

**Jean Meunier**

---

membre du jury

Mémoire accepté le :

# Résumé

Ce mémoire présente une nouvelle approche de segmentation hiérarchique non supervisée d'images naturelles couleur. Notre problématique consiste à extraire de l'image les différents objets présents que l'œil humain peut percevoir en évitant de partitionner l'image en plusieurs régions non significatives (i.e., phénomène de *sur-segmentation*). Le problème de la *sur-segmentation* ne peut être résolu par des algorithmes de segmentation classiques. A cet fin nous avons pensé introduire une méthode hiérarchique basée sur 1) une segmentation Bayésienne-Markovienne qui s'opère au niveau pixel pour former des régions plus au moins homogènes au sens des niveaux de gris, puis de 2) segmenter (fusionner) ces régions obtenues, par partitionnement de graphe d'adjacence de régions pour former d'autres régions plus significatives et plus grandes et homogènes au sens de la texture. Afin que les régions fusionnées convergent véritablement vers les différents objets discernables de l'image, la deuxième phase est répétée plusieurs fois avec un réglage particulier de quelques paramètres de régularisation, jusqu'à la réalisation d'un critère d'arrêt. L'approche a été validée sur la base d'images de Berkeley sur laquelle elle a été quantifiée et comparée aux autres algorithmes existants.

**Mots-clés :** Segmentation Markovienne, segmentation texturale, partition de graphe, fusion de régions, base d'images de Berkeley.

## Abstract

This thesis presents a new approach to hierarchical unsupervised color image segmentation into regions. The problem is how can we extract the objects that the human perceives on the image rather than to partitioning it into several regions not significant (*over-segmentation*). This problem of *over-segmentation* can not be resolved by a classical segmentation algorithm, so we thought to introduce a new hierarchical method based on **1**) Bayesian-Markovian grayscale segmentation, in this step an *over-segmentation* map is obtained with homogeneous regions in gray-level sens, followed by **2**) a region segmentation based on graph partitioning (textural segmentation). Then we have to repeat step 2 (*re-segmentation*) (with automatic adjustment of a regularization parameter) as many times as necessary until a stop criterion is reached, so that from an iteration to another, the regions more similar are merged to form larger and homogeneous objects that represent discernible objects in the picture. The proposed approach was validated on the Berkeley images database.

**Key words :** Markovian segmentation, textural segmentation, graph partitioning, regions merging, image Berkeley database.

# Table des matières

Résumé	i
Table des matières	iii
Liste des figures	vii
Liste des tableaux	x
Liste des abréviations	xi
Remerciements	xii
<b>1 Introduction</b>	<b>1</b>
1.1 Du système visuel à l'image numérique . . . . .	1
1.2 Vision par ordinateur et segmentation d'image . . . . .	2
1.3 Plan général du mémoire . . . . .	3
<b>2 Généralités sur la segmentation</b>	<b>5</b>
2.1 Les différentes approches de segmentation . . . . .	6
2.1.1 Approches utilisant la théorie des graphes . . . . .	6

2.1.1.1	Segmentation par coupe normalisée (NCUT) . . . . .	6
2.1.1.2	Algorithme FH ( <i>Felzenszwalb et Huttenlocher</i> ) . . . . .	7
2.1.2	Approches frontières . . . . .	9
2.1.3	Approches régions . . . . .	9
2.1.3.1	Les méthodes de classification . . . . .	10
2.1.3.2	Méthodes de type croissance de régions . . . . .	11
2.1.3.3	Méthode de type division-fusion . . . . .	11
<b>3</b>	<b>Segmentation Bayésienne non supervisée</b>	<b>13</b>
3.1	Rappel sur les champs de Markov en imagerie . . . . .	13
3.1.1	Notions de systèmes de voisinage et cliques . . . . .	13
3.1.2	Champs de Markov et distribution de Gibbs . . . . .	15
3.1.2.1	Champ de Markov . . . . .	15
3.1.2.2	Champ de Gibbs . . . . .	16
3.1.2.3	Théorème de Hammersley-Clifford . . . . .	17
3.2	Estimation Bayésienne des paramètres . . . . .	17
3.2.1	Estimation MV des données complètes . . . . .	18
3.2.2	Estimateurs des données incomplètes . . . . .	18
3.2.3	Algorithme EM . . . . .	19
3.2.4	Algorithme SEM . . . . .	19
3.2.5	L'algorithme ECI . . . . .	20
3.2.6	L'algorithme des $K$ -moyennes . . . . .	21
3.3	Estimation du champ d'étiquettes . . . . .	22
3.3.1	Estimateur du MAP . . . . .	23

3.3.2	Recuit Simulé (RS) . . . . .	25
3.3.3	Iterated Conditional Modes (ICM) . . . . .	28
<b>4</b>	<b>Segmentation d'images par partitionnement de graphes</b>	<b>30</b>
4.1	Introduction . . . . .	30
4.2	Notions de base sur les graphes . . . . .	33
4.3	Modélisation d'images par un graphe . . . . .	33
4.4	Construction de graphe de régions . . . . .	34
4.4.1	Distance de similarité entre deux régions . . . . .	36
4.4.2	Choix de la distance . . . . .	38
<b>5</b>	<b>Méthodes de partitionnement du graphe de régions</b>	<b>40</b>
5.1	Introduction . . . . .	40
5.2	Bloc-diagonalisation de la matrice d'affinité par les vecteurs propres .	41
5.3	Segmentation par la méthode de flux (MCL) . . . . .	43
<b>6</b>	<b>Re-segmentation</b>	<b>48</b>
6.1	A priori sur le nombre de régions dans la segmentation finale . . . . .	50
<b>7</b>	<b>Evaluation de la segmentation</b>	<b>54</b>
7.1	Introduction . . . . .	54
7.2	Évaluation non-supervisée . . . . .	55
7.3	Évaluation supervisée . . . . .	55
7.4	Vérité-terrain . . . . .	56
7.5	Comparaison de segmentations . . . . .	57



## TABLE DES MATIÈRES

vi

7.6	Critères d'évaluation supervisée . . . . .	59
7.6.1	PRI . . . . .	59
7.6.2	Variation de l'information (VoI) . . . . .	60
7.6.3	Erreur de la cohérence globale (GCE) . . . . .	61
7.6.4	Appariement des frontières (BDE) : . . . . .	63
7.7	Vérification quantitative . . . . .	63
7.8	Influence des paramètres . . . . .	65
7.8.1	Nombre de classes . . . . .	65
7.8.2	Paramètres d'inflation du MCL . . . . .	66
7.8.3	Paramètre de segmentation par les vecteurs propres . . . . .	67
7.8.4	Taille des régions à filtrer avant de lancer le partitionnement par graphe . . . . .	68
7.8.5	Temps de calcul . . . . .	68
8	Discussion et conclusion	70
	Bibliographie	74
A	Résultats 1	i
B	Résultats 2	vi

# Liste des figures

3.1	Relation de voisinage 2D . . . . .	15
3.2	Exemple de cliques pour les relations des 4 ou 8 plus proches voisins .	15
4.1	Diagramme général de l'algorithme proposé : (a) image en couleur, (b) image en niveaux de gris, (c) segmentation au sens des niveaux de gris (ICM, 29 régions), (d) carte et graphe de régions, (e) fusion de régions au sens de la texture (6 régions). . . . .	32
4.2	Graphe non-orienté (à gauche) et graphe orienté (à droite) . . . . .	34
4.3	(a) Graphe non-orienté pondéré, (b) Graphe partitionné en deux sous-graphes disjoints, (c) Matrice d'affinité bloc-diagonalisée . . . . .	35
4.4	(a) Image originale. (b) carte de régions R. (c),(d),(e) parcours de la carte R de gauche à droite/de haut en bas ( $M = 15, N = 5$ ), (f) régions fusionnées (segmentation voulue). . . . .	37
5.1	(a) Ensemble de données, (b) matrice de similarité, (c) matrice de similarité après réorganisation des lignes et des colonnes de la matrice (b), la matrice (c) est bloc-diagonale composée de trois blocs apparents.	43

5.2	Les trois vecteurs propres correspondants aux trois plus grandes valeurs propres (les axes des abscisses correspondent aux données (numéros de régions) et les axes des ordonnées correspondent aux valeurs des vecteurs propres) de la matrice de similarité de l'ensemble des données de la figure 5.1. Les régions 4, 5, 6, 7 et 8 définies par le 1 <sup>er</sup> vecteur propre appartiennent au 1 <sup>er</sup> cluster, les régions 1, 2, 3 et 9 définies par le 2 <sup>ième</sup> vecteur propre appartiennent au 2 <sup>ième</sup> cluster et les régions 10, 11, 12, 13 et 14 définies par le 3 <sup>ième</sup> vecteur propre appartiennent au 3 <sup>ième</sup> cluster. . . . .	43
5.3	Segmentation par la méthode des vecteurs propres EIGEN : de gauche à droite, image originale, ICM, contours ICM, EIGEN ( $s=0.4$ en haut et $s=0.8$ en bas), contours EIGEN, image originale avec contours EIGEN. . . . .	44
5.4	Partitionnement successif du graphe par MCL (source [19]) . . . . .	45
5.5	Segmentation par MCL : de gauche à droite, image originale, ICM, contours ICM, MCL (en haut $r=1.1$ , au milieu $r=1.2$ et $r=1.26$ en bas), contours MCL, image originale avec contours MCL. . . . .	46
6.1	Segmentation MCL : (a) image originale, (b) régions ICM, (c) $MCL_{r=1.5}$ , (d) $MCL_{r=1.25}$ . . . . .	49
6.2	Évolution du paramètre de régularisation $r$ , $\beta_r = 1.4$ , $\tau_r = 25$ . . . . .	50
6.3	Effet du nombre d'itérations sur la segmentation finale : (a) image originale, (b) ICM, (c) $MCL_{nbiter=0}$ (1 itération), (d) $MCL_{nbiter=1}$ (2 itérations). . . . .	51
6.4	Re-segmentation MCL : de gauche à droite, image originale, première itération de segmentation ( $r = 1.5$ ), deuxième itération de segmentation ( $r = 1.3$ ) . . . . .	52
7.1	Quelques images naturelles de la BSDB . . . . .	57

7.2	Segmentation humaine de quelques images de la BSDB . . . . .	58
7.3	Tolérance de mesures d'erreur de segmentation : (a) image originale. (b)-(d) trois segmentations par différents humains. (b) et (d) deux seg- mentations illustrant la division mutuelle. (c) simple division par GCE	62
7.4	Distribution des différentes mesures de performance sur les 300 images de la base de Berkeley pour la configuration : $N_{cls} = 6$ ( <i>nombre de</i> <i>classes</i> ), $F=15 \times 15$ , $\tilde{F}=5 \times 5$ , $\tilde{G}=5 \times 5$ , $nbiter_{max} = 4$ avec $\beta_r = 1.40$ et $\tau_r = 25$ . . . . .	64
7.5	Influence du nombre de classes : à gauche 3 classes, à droite 7 classes	66
7.6	Influence du paramètre d'expansion $r$ : de gauche à droite, image ori- ginale, $MCL_{r=1.2}$ (10 régions), $MCL_{r=1.5}$ (17 régions) . . . . .	67
7.7	Influence du seuil $s$ sur la séparation des composantes des vecteurs propres : de gauche à droite, image originale, $EIGEN_{s=0.4}$ (28 régions), $EIGEN_{s=0.9}$ (21 régions) . . . . .	68
8.1	Effet de camouflage dans l'image couleur sur la segmentation . . . . .	71
8.2	Problème de confusion des objets de l'image lors du passage de la base couleur en carte de niveaux de gris . . . . .	72
8.3	Effet de la segmentation Markovienne sur le processus de regroupement de régions et la segmentation finale : (a) image originale, (b) image en niveaux de gris, (c) carte de segmentation Markovienne (165 régions), (d) carte de segmentation finale par regroupement de régions $MCL_{N_r=4}$ (4 régions) . . . . .	73
8.4	Segmentation sur la base chromatique de saturation : à gauche, seg- mentation ICM de l'image 8.2, à droite, regroupement par MCL. . . . .	73

# Liste des tableaux

3.1	<i>Algorithme ECI</i> . . . . .	21
3.2	<i>Algorithme des k-moyennes</i> . . . . .	23
3.3	<i>Algorithme RS</i> . . . . .	27
3.4	<i>Algorithme ICM</i> . . . . .	29
5.1	<i>Algorithme de la segmentation par les vecteurs propres</i> . . . . .	44
5.2	<i>Algorithme de segmentation par MCL</i> . . . . .	47
6.1	<i>Algorithme de re-segmentation</i> . . . . .	53
7.1	<i>Mesures de performances des méthodes de segmentation. <math>HMC_{[2]}</math> est l'algorithme proposé avec les paramètres : <math>N_{cls} = 6, \beta_r = 1.4, \tau_r = 25, \Gamma = 30, nbiter_{max} = 4, N_r = 75</math>, pour <math>HMC_{[1]}</math>, nous avons utilisé les mêmes paramètres excepté <math>N_r</math>, <math>HMC_{[3]}</math> utilise la méthode des vecteurs propres avec les paramètres : <math>N_{cls} = 6, \beta_s = 1.0, \tau_s = 15, \Gamma = 30, nbiter_{max} = 4, N_r = 75</math></i> . . . . .	65
7.2	<i>Effet du nombre de classes sur la segmentation</i> . . . . .	66

# Listes des abréviations

<b>ASSOC</b>	Associating
<b>BDE</b>	Boundary Displacement Error
<b>BSDb</b>	Berkeley Segmentation Dataset and Benchmark
<b>CUT</b>	Cutting
<b>ECI</b>	Estimation Conditionnelle Itérative
<b>EM</b>	Expectation Maximisation
<b>FH</b>	Felzenszwalb et Huttenlocher
<b>GAR</b>	Graphe d'Adjacence de Régions
<b>GCE</b>	Global Consistency Error
<b>ICM</b>	Iterated Conditional Modes
<b>MAP</b>	Maximum A Posteriori
<b>MCL</b>	Markov Clustering
<b>MPM</b>	Marginal Posterior Mode
<b>MST</b>	Minimal Spanning Trees
<b>MV</b>	Maximum de vraisemblance
<b>NCUT</b>	Normalized Cut
<b>PRI</b>	Probabilistic Rand Index
<b>RS</b>	Recuit simulé
<b>SEM</b>	Stochastic Expectation-Maximisation
<b>VoI</b>	Variation of Information

# Remerciements

Je tiens à adresser mes remerciements à Mr. Max Mignotte, mon directeur de recherche pour son admirable générosité, ses conseils, ses astuces, sa disponibilité, sa tolérance et son soutien tout au long de ces années. Avec perspicacité, il a su choisir les voies les plus valorisantes. Qu'il trouve ici l'expression de ma profonde reconnaissance et de mon admiration.

Je tiens à remercier tous les membres du jury pour m'avoir fait l'honneur d'accepter de juger ce mémoire :

- Jean Meunier
- Phillipe Langlais

Je tiens également à remercier les professeurs, Jean Meunier pour le cours de reconnaissance de formes, Sébastien Roy pour le cours de vision 3D et Neil Stewart pour le cours de modélisation des solides.

Je ne pourrais conclure sans citer et remercier Mr. Said Benameur, chercheur au laboratoire vision qui a été le témoin de mon apprentissage du système Linux et d'autres sujets philosophiques que nous avons eu ainsi que pour son aide à mettre mes premiers pas dans le domaine de l'imagerie.

Mes remerciements par ailleurs s'adressent aussi à tous les membres du laboratoire vision et à tout le personnel du département d'informatique et de recherche opérationnelle.

# Chapitre 1

## Introduction

### 1.1 Du système visuel à l'image numérique

Le système visuel est l'ensemble des organes participants à la perception visuelle, de la rétine au système sensori-moteur. Son rôle est de percevoir et d'interpréter des images. Il est principalement constitué de l'oeil et plus particulièrement la rétine, des nerfs optiques, du chiasma optique, du tractus optique, du corps genouillé latéral, des radiations optiques et du cortex visuel. La lumière entre dans l'oeil où elle est captée par la rétine. Il y a transduction de l'information électromagnétique en potentiels d'actions. L'image est alors analysée au sein même de la rétine en zones de contraste et le résultat de ce traitement est envoyé au reste du système visuel par le nerf optique. Chez les humains, le système visuel est le seul système sensoriel à être directement connecté, via le nerf optique, au cerveau, dû à la nécessité de traiter rapidement l'information visuelle (*wikipedia.com*).

Une image numérique qui représente une scène du monde réel (*image naturelle*) est découpée en une matrice de cellules carrées élémentaires (i.e., indécomposables) et caractérisées par une couleur unique appelée *pixel*. Le traitement de ces pixels (et plus précisément le traitement de la luminance ou de la couleur associée à chacun



d'eux) définie ce qu'on appelle la vision par ordinateur.

## 1.2 Vision par ordinateur et segmentation d'image

David Marr [40] a proposé à la fin des années 70, un paradigme de la vision par ordinateur qui est défini par les quatre points suivants :

1. Reconnaître des contours dans une image permettra de délimiter ses objets et de les structurer.
2. À partir d'une image 2D et de connaissances sur le monde 3D, on peut accéder à la troisième dimension.
3. L'ombre peut être utilisée pour l'extraction du relief.
4. On peut estimer le mouvement des séquences d'images.

À partir de là, il a organisé ses algorithmes en trois niveaux d'abstraction :

1. Le bas-niveau : Les traitements de bas niveau travaillent directement au niveau pixel et nécessitent très peu d'informations sur son contenu.
2. Le niveau intermédiaire : Les algorithmes de ce niveau utilisent les caractéristiques issues du bas-niveau pour former des structures représentant des entités présentes dans l'image (segments, régions, etc.).
3. Le niveau-haut : Dans ce niveau, les algorithmes visent à décrire les objets interprétables de la scène observée.

Ces trois niveaux sont utilisés selon une approche ascendante, où les structures obtenues sont de plus en plus complexes selon le niveau de représentation utilisé.

Le cadre général dans lequel s'inscrit ce mémoire est celui de la segmentation d'images couleur. Cette opération de segmentation, comme pour le cas de l'humain qui voit généralement des objets physiques ou des régions à partir desquelles il est capable d'extraire ou de différencier les objets contenus dans cette images (par l'utilisation des connaissances de haut niveaux), vise à modéliser cette opération en utilisant

une machine numérique tel qu'un ordinateur. Cette modélisation est un processus de synthèse qui consiste à extraire les caractéristiques géométriques d'une image en se basant sur la richesse de l'information que l'image comporte telle que la nuance de la couleur, la texture, les reflets de la transparence ou autres éléments qui peuvent être jugés utiles pour une telle opération. La segmentation d'image est alors, définie comme une opération de traitement d'images qui a pour but de regrouper des pixels entre eux suivant des critères prédéfinis pour former des régions homogènes.

Ce mémoire décrit le résultat de la méthode de segmentation *non-supervisée* d'images naturelles couleur que nous avons proposé. Il porte sur la segmentation d'images couleur en régions qui est une étape fondamentale du traitement d'image et de la vision par ordinateur. Nous avons été motivés par une problématique intéressante visant à délimiter réellement (ou à extraire) les objets que l'oeil humain perçoit sur l'image à segmenter plutôt que de partitionner l'image en plusieurs régions non significatives (i.e., trop de subdivisions dans l'image segmentée). A cet effet, nous avons constaté que ce problème de régions non significatives (appelé *sur-segmentation*) ne peut pas être résolu par un algorithme de segmentation classique. C'est pourquoi nous avons pensé à introduire une méthode originale basée sur l'application de la segmentation Bayésienne par une approche Markovienne qui s'effectue au niveau pixel (au sens des niveaux de gris) suivie par une méthode de niveau supérieur s'effectuant au niveau région (au sens de la texture) en utilisant la segmentation par partition de graphe de régions puis en réitérant ce même processus autant de fois que nécessaire jusqu'à la réalisation d'un critère d'arrêt.

### 1.3 Plan général du mémoire

Ce mémoire est décomposé en 9 chapitres. Le chapitre 2 consiste à présenter une brève définition de la segmentation d'image numérique ainsi que les différentes méthodes habituellement utilisées pour réaliser cette opération. Dans le chapitre 3 nous abordons le problème de la segmentation Bayésienne par une approche Markovienne

et nous décrivons l'estimation des paramètres du modèle Markovien défini pour une segmentation *non-supervisée* au sens des niveaux de gris des images naturelles. Dans le chapitre 4, nous traitons de la modélisation de l'image *sur-segmentée* par un graphe et son utilisation dans le domaine de la segmentation par partitionnement du graphe. Nous décrivons ainsi deux méthodes pour réaliser le partitionnement du graphe. La première méthode est l'utilisation des vecteurs propres qui sera étudiée dans le chapitre 5 et la deuxième méthode est résumée par la méthode de flux dans des graphes appelée le MCL (Markov CLustering), cette méthode fait l'objet du chapitre 5. Le phénomène de la *sur-segmentation* (plusieurs petites régions non significatives dans la segmentation finale) produit par la segmentation Markovienne reste encore à ce niveau un problème, même avec l'utilisation des méthodes basées sur les graphes. A cet effet, nous avons donc pensé à réitérer plusieurs fois le processus de segmentation par la méthode de partitionnement de graphes jusqu'à la réalisation d'un critère d'arrêt. La réitération du processus de segmentation qui sera appelée par la suite la *re-segmentation*, dépend d'une variation de quelques paramètres ajustables (décrits dans le chapitre 5) au cours des itérations. L'algorithme de la *re-segmentation* sera étudié dans le chapitre 6. Le chapitre 7 est consacré au problème de l'influence des paramètres et l'évaluation des résultats des méthodes de la segmentation introduites dans le chapitre 5. Enfin une conclusion fera l'objet du chapitre 8 .

## Chapitre 2

# Généralités sur la segmentation

Dans ce chapitre, nous abordons le problème général de la segmentation d'images en couleur texturées qui consiste à trouver les régions homogènes et les contours pertinents des différents objets discernables et significatifs dans l'image.

**Définition 1 :** *La segmentation d'image est définie comme étant une partition de l'image  $I$  en  $n$  sous ensembles  $R_i$ , appelés régions, tels que  $\forall i, j \in \{1, \dots, n\}$  :*

1.  $I = \bigcup_{i=1}^n R_i$
2.  $R_i \neq \emptyset$
3.  $i \neq j, R_i \cap R_j = \emptyset$

Où  $\bigcup$  représente une union d'ensembles disjoints et  $\cap$  représente l'intersection ensembliste. Ces sous-ensembles  $R_i$  constituent les différentes régions de l'image. Une segmentation d'image est donc sa décomposition en un ensemble de régions homogènes où (a) chaque pixel appartient à une région et une seule, (b) tous les pixels doivent être traités, (c) toute région doit être connexe.

## 2.1 Les différentes approches de segmentation

Le problème de la segmentation d'images peut être résolu par différentes approches. La méthode de segmentation que nous proposons dans ce mémoire fait partie des approches de segmentation en régions et se base sur une classification Bayésienne (approche Markovienne) suivie par une analyse de graphe qui fusionnera certaines régions de cette segmentation.

### 2.1.1 Approches utilisant la théorie des graphes

Les approches utilisant la théorie des graphes, consistent à créer un graphe à partir de l'image et d'exploiter les différentes propriétés de cette théorie pour trouver les différentes régions de l'image. L'idée générale est de modéliser les pixels par les nœuds de ce graphe et les caractéristiques (distance, similarité, ...) reliant ces pixels par des arcs. Parmi les méthodes clés impliquées, on trouve la méthode de coupe minimale normalisée du graphe (NCUT) [62][64], et la méthode de flux (MCL) [19].

#### 2.1.1.1 Segmentation par coupe normalisée (NCUT)

Dans la segmentation par coupe normalisée, l'image est considérée comme un graphe pondéré complet non-orienté  $G = (V, E)$  avec  $V$  l'ensemble des nœuds et  $E$  l'ensemble des arcs reliant ces nœuds. La segmentation d'une image revient à partitionner le graphe correspondant en réalisant des coupures qui minimisent un certain critère (coût minimal). Plus précisément, les pixels dans l'image sont considérés comme des nœuds avec des arcs reliant ces différents pixels. La matrice de poids  $W$  est construite tel que  $\Omega(i, j)$  est le poids entre le pixel  $i$  et le pixel  $j$ . Le graphe est ainsi coupé en deux sous-graphes  $A$  et  $B$  où

$$A \cup B = G \text{ et } A \cap B = \emptyset$$

La coupure normalisée ( $NCUT$ ), pour le problème de définition d'une partition de  $V$  en deux *clusters*  $A$  et  $B$ , s'écrit :

$$NCUT(A, B) = \frac{CUT(A, B)}{ASSOC(A, V)} + \frac{CUT(A, B)}{ASSOC(B, V)} \quad (2.1)$$

Avec,  $ASSOC(A, V)$  est définie plus précisément comme étant la somme des mesures des arcs entre les nœuds du *cluster*  $A$  et tous les autres nœuds de  $V$  (les nœuds de  $A$  inclus par conséquent).

$$ASSOC(A, V) = \sum_{i \in A} \sum_{j \in V} \Omega_{ij} \quad (2.2)$$

et  $CUT(A, B)$  est définie comme étant la somme des mesures des arcs reliant les nœuds de  $A$  et  $B$ , ou encore la somme des mesures des arcs que l'on enlèverait si on devait séparer les deux clusters  $A$  et  $B$ .

$$CUT(A, B) = \sum_{i \in A} \sum_{j \in \{V-A\}} \Omega_{ij} \quad (2.3)$$

Cette définition normalisée de la coupure a été largement discuté dans l'article [64].

### 2.1.1.2 Algorithme FH (*Felzenszwalb et Huttenlocher*)

Les auteurs dans [22] se servent d'une approche basée sur les graphes pour segmenter une image. Ils considèrent un graphe non orienté  $G = (V, E)$  avec  $v_i \in V$ , l'ensemble des éléments (nœuds) à segmenter, et  $(v_i, v_j) \in E$  les paires des nœuds voisins (arcs). Chaque arc  $(v_i, v_j) \in E$  a un poids non-négatif qui lui correspond  $w((v_i, v_j))$  mesurant la dissimilarité entre les nœuds voisins  $v_i$  et  $v_j$ . Dans le cas d'une segmentation d'images les éléments dans  $V$  sont les pixels de l'image et le poids d'un

arc est une mesure de dissimilarité entre les deux pixels connectés par cet arc (i.e., la différence en intensité, couleur, texture ou autre attribut local).

Dans cette approche, une segmentation  $S$  est la partition de  $V$  en plusieurs composantes (ou régions) telle que chaque région  $C \in S$  correspond à une composante connectée dans le graphe  $G' = (V, E')$ , où  $E' \subseteq E$ .

Le but de cette approche est d'avoir des éléments beaucoup plus similaires dans une même région et plus différents s'ils appartiennent à des régions différentes. Cela signifie que les arcs entre deux nœuds dans la même région devraient être relativement de faibles poids, et les arcs entre les nœuds de différentes composantes devraient avoir des poids forts.

Les auteurs de cette approche ont défini un prédicat  $D$ , pour évaluer s'il y a, ou non, une frontière évidente entre deux régions dans la segmentation. Ce prédicat est basé sur une mesure de la similarité entre les éléments le long de la frontière séparant deux régions relativement à une mesure de dissimilarité entre des pixels voisins dans chacune de ces régions. Ce prédicat compare la différence inter-composantes (inter-régions) à la différence intra-composantes (à l'intérieur de chaque région) en respectant certaines caractéristiques locales des données.

La *différence interne* d'une composante  $C \subseteq V$  est définie comme le poids le plus fort dans l'arbre de recouvrement minimal  $MST(C, E)$  :

$$Int(C) = \max_{e \in MST(C, E)} w(e). \quad (2.4)$$

Une autre *différence* entre deux composantes  $C_1$  et  $C_2 \subseteq V$ , est aussi considérée pour être le poids le plus faible de l'arc connectant ces deux composantes, c'est :

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j). \quad (2.5)$$

Le prédicat  $D$ , se définit comme suit :

$$D(C_1, C_2) = \begin{cases} \text{Vrai} & \text{si } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{Faux} & \text{ailleurs} \end{cases} \quad (2.6)$$

Où la différence interne minimal  $MInt$ , est défini comme,

$$MInt(C_1, C_2) = \min \left( Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2) \right). \quad (2.7)$$

Le seuil  $\tau$  contrôle le degré pour lequel la similarité intra-régions doit être grande que la similarité inter-régions, dans l'ordre que la frontière entre ces deux régions soit évidente ( $D$  est vrai). Pour plus détails voir l'article [22].

### 2.1.2 Approches frontières

Les approches frontières font partie des méthodes les plus classiques en segmentation d'images. Ce type d'approches consiste à extraire des primitives définies par les lignes de discontinuité séparants les régions qui ne diffèrent entre elles que par leurs niveaux de couleur ou par leurs textures. Pratiquement, il s'agit de mettre en évidence les zones de transition et de détecter les différentes frontières qui séparent les régions dans une image. Les approches frontières elles même, peuvent être classées en plusieurs catégories [12]. On peut distinguer normalement les modèles dérivatifs [16][38][44], surfaciques [28][31] et variationnelles [20][61][45].

### 2.1.3 Approches régions

Les approches régions sont des techniques de segmentation qui consistent à identifier et à localiser les ensembles connexes de pixels. Autrement-dit, contrairement



aux approches frontières qui recherchent des dissimilarités, les approches régions recherchent plutôt la similarité. Ces approches produisent des régions fermées. Les approches régions peuvent être classées en trois types de méthodes :

### 2.1.3.1 Les méthodes de classification

Ces méthodes ont pour but de partitionner l'image en plusieurs classes. Chaque pixel est attribué à une et une seule classe. Les méthodes de classification sont séparées en :

1. Méthodes probabilistes, parmi celles ci, on trouve les méthodes de mélange de lois [15][8][9], Markovienne [26], les machines à vecteurs de support [69].
2. Méthodes déterministes : On trouve dans ce type de méthodes les réseaux de neurones [50][32], k-moyennes [17][54][35], c-moyennes floues [25][30][1], mean-shift [14].

#### Algorithme de segmentation basé sur le Mean-Shift

A l'origine, le Mean-shift (proposé par *Fukunaga* en 1975) est une procédure itérative (non-paramétrique) d'ascension de gradient, utilisée pour estimer les modes (i.e., le maximum local) d'une densité de probabilité associée à une distribution de points. Cette procédure a été utilisée pour la première fois en 1997 dans le cadre de la segmentation d'images par *comaniciu et al.* [14]. Dans cette application du Mean-shift à la segmentation, le nuage de points considéré est l'ensemble des vecteurs à cinq dimensions indépendantes (2 pour la position et 3 dimensions pour la couleur) associées à chaque pixel de l'image.

Le principe de la segmentation par Mean-shift consiste à trouver pour chacun de ses pixels, son mode ou maximum local le plus proche. Plus précisément, cette procédure consiste pour chaque pixel  $p$  :

1. à chercher l'ensemble  $E$  des points qui sont dans son voisinage.
2. à déplacer  $p$  par une translation qui le conduira vers l'isobarycentre de l'ensemble  $E$ .
3. à répéter le processus depuis l'étape 1 jusqu'à convergence.

Les déplacements successifs vers l'isobarycentre font converger le pixel  $p$  vers les zones de fortes densités, i.e., le mode. La classification ou la segmentation finale de l'ensemble des pixels de l'image est ensuite obtenue en regroupant dans une même classe tous les points ayant convergé vers le même mode.

#### 2.1.3.2 Méthodes de type croissance de régions

Ces méthodes consistent à faire croître chaque région autour d'un pixel de départ appelé un germe. Les régions sont créées les unes après les autres suivant deux phases : La phase d'initialisation est la phase du choix d'un nouveau germe. La phase itérative dans laquelle les pixels voisins sont agrégés au germe selon un critère d'homogénéité jusqu'à convergence [70][29][12].

#### 2.1.3.3 Méthode de type division-fusion

Ces méthodes consistent à découper (division) itérativement l'image jusqu'à l'obtention de blocs homogènes selon un critère donné. Puis regrouper les blocs voisins de sorte que le bloc résultant respecte un critère d'homogénéité. Ces méthodes peuvent faire appel à la théorie des graphes, au partitionnement de Voronoï [12], à une structure de données de type arbre quaternaire [67][60] ou aux approches pyramidales [43].

#### Remarque :

Une différence entre la première méthode et les deux dernières, vient du fait que si par

exemple, un objet existe en plusieurs endroits séparés dans l'image, il seront identifiés comme des objets différents par les deux dernières méthodes, tandis qu'il peuvent faire part d'un autre objet ayant les mêmes caractéristiques (classe) dans le cas d'une segmentation par la première méthode.

## Chapitre 3

# Segmentation Bayésienne non supervisée

Dans une image réelle, les pixels voisins ont souvent un niveau de gris similaire. Dans un cadre probabiliste, une telle régularité peut être modélisée par les champs de Markov et être utilisée avec succès dans l'analyse d'image, la vision par ordinateur [26][27][33][59] et plus généralement pour tout problème inverse mal posés dans un cadre Bayésien.

### 3.1 Rappel sur les champs de Markov en imagerie

#### 3.1.1 Notions de systèmes de voisinage et cliques

Définir un champ de Markov ou un champ de Gibbs revient à introduire quelques notions de la théorie des graphes. Nous allons limiter notre étude à décrire quelques notions fondamentales dans le cadre du traitement et l'analyse d'images. Dans celui-ci, l'image, peut être considérée comme une grille bidimensionnelle de sites (ou pixels)  $S = [1, m] \times [1, n]$ , dans laquelle chaque site  $s \in S$  peut être défini soit par :

- Sa coordonnée  $\{(i, j), i = 1, \dots, m, j = 1, \dots, n\}$
- Son numéro d'ordre  $S = \{s_1, s_2, \dots, s_N\}$ ,  $N = n \times m$  est le cardinal de  $S$ .

La connexion des sites des plus proches voisins sur cette grille  $S$ , peut être modélisée par un graphe non orienté  $G = \{S, V(S)\}$ , où  $V(S) = \{v_s(S); s \in S\}$  est appelé système de voisinage qui possède les propriétés suivantes :

$$\left\{ \begin{array}{l} (i) \quad \forall s \in S, s \notin v_s(S) \\ \text{et} \\ (ii) \quad \forall \{s, t\} \subset S, s \in v_t(S) \leftrightarrow t \in v_s(S) \end{array} \right. \quad (3.1)$$

Un sous ensemble  $c$  de  $S$  est appelé une clique relative au système de voisinage  $V$  si :

$$\left\{ \begin{array}{l} \exists s \in S : c = \{s\} \\ \text{ou} \\ \forall \{s, t\} \subset c, t \in v_s(S) \end{array} \right. \quad (3.2)$$

Le sous ensemble  $c$  est un singleton et les parties de  $S$  sont les éléments voisins deux à deux.

On utilise très souvent la relation des 4-plus proches voisins, ou celles des 8-plus proches voisins dans le cas bidimensionnelle (cf. figure 3.1). A partir des systèmes de voisinage, on peut décrire les graphes de voisinage reliant simplement deux sites  $s$  et  $t$  si ils sont voisins. Un exemple de cliques est donné à la figure 3.2.

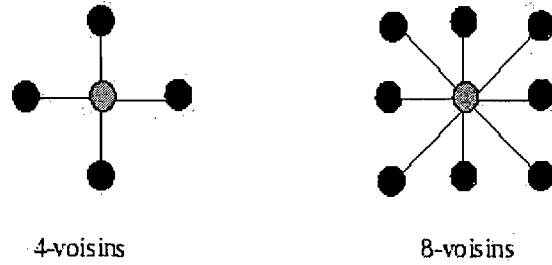


FIG. 3.1 – Relation de voisinage 2D



FIG. 3.2 – Exemple de cliques pour les relations des 4 ou 8 plus proches voisins

### 3.1.2 Champs de Markov et distribution de Gibbs

Étant donné un ensemble  $S$  fini de sites, on s'intéresse à des ensembles  $\{x_s; s \in S\}$  de variables indexées par les éléments de  $S$  prenant ses valeurs dans l'ensemble  $\mathbb{E} = \{e_1, \dots, e_k\}$ . Dans le cas de la segmentation d'images;  $k$  est le nombre de classes et  $\mathbb{E}$  l'ensemble des étiquettes possibles avec  $\Omega = \underbrace{\mathbb{E} \times \mathbb{E}, \dots \times \mathbb{E}}_N$ , l'espace des configurations possibles (fini dans notre cas).

#### 3.1.2.1 Champ de Markov

On définit un champ aléatoire  $X$  assimilé au vecteur aléatoire  $\{X_s; s \in S\}$  indexé par les éléments de  $S$  comme étant un champ Markovien sur un graphe  $\{S, v(S)\}$  si et seulement si il vérifie

- (i)  $\forall x \in \mathbb{E}, P_X(x) > 0$  (contrainte de positivité)
- (ii)  $\forall s \in S, \forall x \in \mathbb{E}$ , alors

$$P_{X_s|X_t}(x_s|x_t, t \in S, t \neq s) = P_{X_s|X_t}(x_s|x_t, t \in v_s(S)) \quad (3.3)$$

la propriété 3.3 est appelée propriété de *Markoviennité*, qui dit que le changement d'état apporté à un site est entièrement dépendant de l'état de ses pixels voisins. Nous pouvons constater que l'intérêt porté aux champs de Markov réside essentiellement dans la notion de la localité (voisinage le plus proche) [52][49].

### 3.1.2.2 Champ de Gibbs

On dit aussi que  $X$ , champ de Gibbs sur un graphe de voisinage  $\{S, v(S)\}$ , est la distribution statistique  $P_x(X)$  donnée par :

$$\forall x \in \mathbb{E}, P_X(x) = \frac{1}{Z} \exp\left[-\frac{U(x)}{T}\right] \quad (3.4)$$

où

$$Z = \sum_{x \in \mathbb{E}} \exp\left[-\frac{U(x)}{T}\right] \quad (3.5)$$

$Z$  est une constante de normalisation appelée fonction de partition et  $U(x)$  est la fonction d'énergie :

$$U(x) = \sum_{c \in C} V_c(x) \quad (3.6)$$

$V_c(x)$  est appelée la fonction de potentiels sur les cliques du graphe  $\{S, v(S)\}$  et  $C$ , l'ensemble des cliques défini sur notre voisinage.

### 3.1.2.3 Théorème de Hammersley-Clifford

Le théorème de Hammersley-Clifford [3] établit qu'un champ de Markov est un champ de Gibbs, étant considéré que le premier se caractérise par sa propriété locale et le deuxième par sa propriété globale.

## 3.2 Estimation Bayésienne des paramètres

Une modélisation probabiliste pour le problème de la segmentation consiste donc à [74] :

1. définir un champs de Gibbs  $P_X(x)$  pour les étiquettes  $x$ .
2. construire un modèle de formation de l'image.

Dans cette étape on se pose la question : *Que devrait être l'image  $y$  étant donné les étiquettes  $x$  ?* Mathématiquement c'est la loi statistique de  $y$  étant donné  $x$  ou  $P_{Y|X}(y|x)$ .

A partir des étapes 1 et 2, on obtient un modèle conjoint pour le couple  $(x, y)$  :

$$P_{X,Y}(x, y) = P_X(x)P_{Y|X}(y|x) \quad (3.7)$$

3. Estimation des paramètres : pratiquement dans les applications réelles, les paramètres sont souvent inconnus. Il faut donc les estimer à partir de l'image observée. D'un point de vue statistique, le problème d'estimer les paramètres de la loi  $P_{Y|X}(y|x)$  est équivalent au problème de l'estimation des paramètres d'un mélange de distributions. Si nous avons une réalisation du champ des étiquettes (une réalisation de  $X$  est connue), la tâche est alors relativement facile,



il existe plusieurs méthodes standards telle que la méthode du maximum de vraisemblance (MV) sur les données complètes.

### 3.2.1 Estimation MV des données complètes

Étant donné un champ d'étiquettes  $X = (x_1, x_2, \dots, x_N)$  connu et un champ des observations  $Y = (Y_1, Y_2, \dots, Y_N)$  ( $N$  étant le nombre de pixels de l'image) comme variables aléatoires indépendantes et régies par une loi statistique  $P_{Y|\Phi_y}(y|\Phi_y)$  du paramètre  $\Phi_y$ . Soit  $y = (y_1, y_2, \dots, y_N)$ , une configuration particulière de  $Y$ . L'estimation de  $\Phi_y$  au sens du MV consiste à trouver le paramètre  $\Phi_y$  tel que

$$\hat{\Phi}_y = \arg \max_{\Phi_y} \{\ln P_{Y|\Phi_y}(y|\Phi_y)\} \quad (3.8)$$

**Loi gaussienne :**

Lorsque  $P_{Y|\Phi_y}(y|\Phi_y)$  est une loi gaussienne  $\mathcal{N}(y; \Phi_y)$  de paramètre  $\Phi_y = (\mu, \sigma^2)$  (moyenne et variance), l'estimateur MV des données complètes s'écrit

$$\hat{\mu}_{MV} = \frac{1}{N} \sum_{i=1}^N y_i \quad (3.9)$$

$$\hat{\sigma}_{MV}^2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \hat{\mu}_{MV})^2 \quad (3.10)$$

### 3.2.2 Estimateurs des données incomplètes

Dans les applications réelles, nous devons estimer les paramètres à partir des données non-étiquetées (une réalisation de  $X$  est inconnue). Plusieurs méthodes ont été proposées pour résoudre ce problème, EM (*Expectation-Maximisation*) [15], sa version stochastique le SEM, et l'ECI (*Estimation Conditionnelle Itérative*) [55].

### 3.2.3 Algorithme EM

L'algorithme EM [2] est un algorithme itératif très utilisé pour la recherche des paramètres au sens du maximum de vraisemblance sur des données incomplètes. L'astuce de l'EM est de ne pas maximiser la fonction de vraisemblance intractable mais une fonction plus simple définie par la densité des données complètes :  $P_{Z|\Phi_y}(z|\Phi_y)$ , avec  $Z = (X, Y)$  ( $Y$  est la variable cachée). Chaque itération de l'EM est composée de deux étapes, la première est l'étape de calcul d'espérance et la deuxième est l'étape de maximisation des données estimées.

- Étape E (Estimation)

On calcule la fonction  $\mathcal{Q}(\Phi_y, \Phi_y^{[p]})$

$$\mathcal{Q}(\Phi_y, \Phi_y^{[p]}) \triangleq E[\ln P_{Z|\Phi_y}(z|\Phi_y)|y, \Phi_y^{[p]}] \quad (3.11)$$

- Étape M (Maximisation)

On détermine  $\mathcal{Q}^{[p+1]}$  qui maximise  $\mathcal{Q}(\Phi_y, \Phi_y^{[p]})$

$$\mathcal{Q}^{[p+1]} = \arg \max_{\Phi} \mathcal{Q}(\Phi_y, \Phi_y^{[p]}) \quad (3.12)$$

### 3.2.4 Algorithme SEM

L'algorithme SEM (*Stochastique EM*) a pour but de déterminer les composants d'un mélange de probabilité, ainsi que le nombre lui-même de ces composantes, par une approche d'apprentissage probabiliste. Contrairement à l'algorithme EM l'algorithme SEM réduit le risque de tomber dans un minimum local de l'énergie de vraisemblance [24]. A cette fin les auteurs de cet algorithme ont proposé d'intercaler une étape stochastique de classification entre les étapes E et M. Il s'agit d'un étiquetage des pixels selon leur distribution conditionnelle courante.

### 3.2.5 L'algorithme ECI

Contrairement aux algorithmes précédents qui sont des algorithmes locaux, l'algorithme ECI [55] est un algorithme Markovien d'estimation des paramètres dans le cas de données cachées. Soient  $X$  et  $Y$  deux champs aléatoires dont la loi conjointe dépend d'un paramètre  $\Phi_y$ , et soit  $\hat{\Phi}_y = \hat{\Phi}(X, Y)$  un estimateur MV de celui-ci défini à partir des données complètes. Le champ  $X$  étant non-observable, l'idée de l'ECI est d'approximer  $\hat{\Phi}_y(X, Y)$  au sens de l'erreur quadratique moyenne par l'espérance conditionnelle. Or, l'espérance conditionnelle de  $\hat{\Phi}_y$  dépend du paramètre  $\Phi_y$ . Ainsi, pour calculer  $E[\hat{\Phi}_y(X, Y)|Y]$ , l'approche itérative suivante a été proposée :  $\hat{\Phi}^{[p+1]} = E[\hat{\Phi}_y(X, Y)|Y = y]$  avec  $E$  l'espérance conditionnelle relative au paramètre  $\hat{\Phi}_y^{[p]}$  obtenu à l'itération  $p$ . La procédure itérative ECI se définit comme suit

1. Initialiser le vecteur de paramètres  $\hat{\Phi}_y^{[0]}$ .
2. Calculer en chaque itération  $\hat{\Phi}_y^{[p+1]}$  à partir de  $\hat{\Phi}_y^{[p]}$  et  $Y = y$  par :

$$\hat{\Phi}_y^{[p+1]} = E_p[\hat{\Phi}_y(X, Y)|Y = y] = \sum_x \hat{\Phi}_y(x, y)P(x|y, \hat{\Phi}_y^{[p]}) \quad (3.13)$$

Lorsque  $E_p[\hat{\Phi}_y(X, Y)|Y = y]$  n'est pas explicitement calculable, mais la simulation des réalisations de  $X$  selon la loi *a posteriori* est possible, on utilise l'approximation

$$\hat{\Phi}_y^{[p+1]} = \frac{1}{n}[\hat{\Phi}_y(x^{[1]}, y) + \dots + \hat{\Phi}_y(x^{[n]}, y)] \quad (3.14)$$

Où  $x^{[1]}, \dots, x^{[n]}$  sont les  $n$  simulations des réalisations de  $X$  selon la distribution *a posteriori*  $P_{X|Y, \Phi}(x|y, \hat{\Phi}^{[p]})$ .

□ **Algorithme** : Estimation Conditionnelle Itérative (ECI)

└  $\hat{\Phi}_y^{[0]}$  Vecteur de paramètre initial

- Calculer  $\hat{\Phi}_y^{[p+1]}$  à partir de  $\hat{\Phi}_y^{[p]}$ 
  1. Simulation de  $n$  réalisations  $(x^{[1]}, \dots, x^{[n]})$  de  $X$  selon la distribution *a posteriori*  $P_{X|Y,\Phi}(x|y, \hat{\Phi}_y^{[p]})$  grâce à l'échantillonneur de Gibbs et sur la base de  $\hat{\Phi}_y^{[p]}$ .
  2. On utilise les estimateurs MV des données complètes sur ces différentes images segmentées  $x^{[i]}$  et on calcule  $\hat{\Phi}_y^{[p+1]}$  par la relation
$$\hat{\Phi}_y^{[p+1]} = \frac{1}{n} [\hat{\Phi}_y(x^{[1]}, y) + \dots + \hat{\Phi}_y(x^{[n]}, y)]$$
  3. Si  $\hat{\Phi}_y^{[p+1]} \not\approx \hat{\Phi}_y^{[p]}$   $p \leftarrow (p + 1)$  retourner à (1)

TAB. 3.1 – *Algorithme ECI*

### 3.2.6 L'algorithme des $K$ -moyennes

L'algorithme des  $k$ -moyennes [39] est une méthode de classification automatique qui a pour objectif de partitionner l'espace des attributs en  $K$  classes. Le principe consiste à partir d'une partition initiale, d'améliorer itérativement la partition de l'espace en minimisant la variance intraclasse (i.e., la dispersion entre les classes). L'algorithme des  $k$ -moyennes est nécessaire comme étape d'initialisation des algorithmes (**EM**, **SEM**, **ECI**). Ces algorithmes convergent d'autant plus vite que l'initialisation des paramètres est proche des valeurs recherchées. La répartition des différents attributs  $x_l$  dans les différents regroupements est effectuée de façon à minimiser un indice de dispersion. Cet indice de dispersion s'écrit :

$$J = \sum_{i=1}^K \sum_{x_l \in C_i} |x_l - c_i|^2 \quad (3.15)$$

avec  $c_i$  le centre du regroupement  $C_i$  et la deuxième sommation se fait sur tous les attributs  $x_l$  appartenant à ce regroupement.

L'algorithme ainsi défini converge vers un minimum local, qui se traduit par une partition de l'espace des données en des classes séparées par des hyperplans et réalise une tessellation de Voronoï basée sur les noyaux des clusters. La qualité de la solution ainsi trouvée dépend des noyaux initiaux. De plus la sensibilité de l'algorithme à l'initialisation est d'autant plus grande que la dimensionnalité des données est grande [6].

### 3.3 Estimation du champ d'étiquettes

Segmenter une image revient à estimer un ensemble de variables aléatoires  $X$  cachées, à partir d'observation  $Y = y$ . Ainsi, le problème est de déterminer une estimation  $\hat{x} \in \Omega$  de  $x$  à partir de  $y$ , obtenue par l'optimisation d'un critère. Dans le cas d'une segmentation Bayésienne l'estimateur Bayésien est alors obtenu en minimisant l'espérance d'une fonction de coût conditionnellement aux observations :

$$\hat{x} = \arg \min_x E[C(X, x) | Y = y] \quad (3.19)$$

où  $X$  est un vecteur aléatoire représentant la solution exacte et  $C(X, x)$  représente le coût de choisir  $x$  alors que la solution exacte est  $X$ . A chaque fonction de coût correspond ainsi un estimateur Bayésien. On fait référence, entre autre, l'estimateur du *maximum a posteriori* (**MAP**) et l'estimateur du *mode des marginales a posteriori* (**MPM**).

□ **Algorithme : k-moyennes**

1. On choisit les  $K$  premiers attributs comme étant les  $K$  centres  $(c_1^{[1]}, \dots, c_k^{[1]})$  des  $K$  regroupements

$$c_i^{[1]} = x_i \quad 1 \leq i \leq K \quad (3.16)$$

2. A l'itération  $k$ , on associe l'attribut  $x_l$ , ( $1 \leq l \leq M$ ) au regroupement  $C_i$  (noté  $C_i^{[p]}$ ) si :

$$\|x_l - c_i^{[p]}\| < \|x_l - c_j^{[p]}\| \quad \forall j \neq i \quad (3.17)$$

En fait, on associe l'attribut  $x_l$  au regroupement dont le centre lui est le plus proche.

En cas d'égalité, on associe  $x_l$  arbitrairement au regroupement  $i$  ou  $j$ .

3.  $C_i^{[p]}$  correspond au regroupement  $i$  constitué à l'itération  $p$  après l'étape 2. On détermine le nouveau centre de chaque regroupement par :

$$c_i^{[p+1]} = \frac{1}{N_i} \sum_{x \in C_i^{[p]}} x \quad \forall i \quad (3.18)$$

Avec  $N_i$  le nombre d'attributs de  $C_i^{[p]}$ .

4. Si les centres des regroupements à l'itération  $p$  sont les mêmes que ceux obtenus à l'itération précédente, alors les regroupements constitués à l'itération  $p$  et  $(p-1)$  sont identiques et l'algorithme a convergé, sinon on retourne à l'étape 2.

Si  $c_i^{[p+1]} = c_i^{[p]} \quad \forall i$  fin de l'algorithme sinon on retourne en 2.

TAB. 3.2 – Algorithme des k-moyennes

### 3.3.1 Estimateur du MAP

Le **MAP** est associé à la fonction de coût suivante :

$$C(X, x) = 1 - \delta_X(x) \quad (3.20)$$

Où  $\delta_X(x)$  est la masse de Dirac en  $X$  définie par  $\delta(x_s, x_t) = 0$  pour  $x_s \neq x_t$  et  $\delta(x_s, x_t) = 1$  pour  $x_s = x_t$ . Cette fonction de coût est sévère, car elle pénalise toute différence entre deux configurations de manière identique. L'équation (3.19) devient :

$$\begin{aligned}\hat{x}_{MAP} &= \arg \min_x \{ \sum_{x' \in E} C(X, x') \cdot P_{X|Y}(x' | y) \} \\ &= \arg \min_x \{ 1 - P_{X|Y}(x | y) \} \\ &= \arg \max_x \{ P_{X|Y}(x | y) \}\end{aligned}\quad (3.21)$$

L'estimation au sens du MAP, revient donc à maximiser la probabilité *a posteriori*. L'existence de solution pour cette maximisation ne fait aucun doute, puisque l'espace de configuration est fini. Malgré cela, elle ne peut pas être réalisée directement car cet espace de configuration est trop grand. Par exemple pour une image de  $256 \times 256$  pixels avec des étiquettes binaires pour  $x_s$ , le calcul direct nécessite le calcul de  $2^{256 \times 256}$  probabilités différentes. Malgré tout, l'un des intérêts de ce critère est qu'on peut lui appliquer la règle de Bayes, ce qui donne après simplification :

$$\hat{x}_{MAP} = \arg \max_x \{ P_X(x) \cdot P_{Y|X}(y | x) \} \quad (3.22)$$

Si on note  $P_{Y|X}(y | x) \triangleq \exp\{-V(x, y)\}$ , et en vertu du théorème de Hammersley-Clifford,  $X$  est un champ de Markov et suit une loi de Gibbs :

$$P_X(x) = \frac{1}{Z} \exp\{-U(x)\} \quad (3.23)$$

avec  $U(x)$ , somme des potentiels locaux sur les cliques  $c \in C$ , nous avons :

$$\hat{x}_{MAP} = \arg \max_x \left\{ \frac{1}{Z} \exp\{-U(x)\} \cdot \exp\{-V(x, y)\} \right\} \quad (3.24)$$

$$= \arg \max_x \left\{ \frac{1}{Z} \exp\{-(U(x) + V(x, y))\} \right\} \quad (3.25)$$

$$= \arg \min_x \{ U(x) + V(x, y) \} \quad (3.26)$$

La recherche d'un tel minimum est ensuite un problème classique d'optimisation combinatoire. L'utilisation des méthodes classiques n'est pas possible à cause de la non-convexité de la fonction d'énergie. L'idée de la solution vient de la physique statistique. En 1953, *Metropolis et al.* ont proposé une simulation *Monte-Carlo* pour trouver les états d'équilibres des systèmes thermodynamiques. Dans les années 80, *Kirkpatrick et al.* ont montré l'analogie entre la minimisation d'une fonction non-convexe et l'état d'équilibre des systèmes thermodynamiques. Ils ont fait l'analogie entre l'état d'équilibre d'un système thermodynamique et une fonction de coût à minimiser avec exécution de l'algorithme de *Metropolis* en utilisant une séquence de températures décroissant lentement. Ils ont appelé ce nouvel algorithme *le recuit simulé* [34].

Malheureusement les algorithmes de recuit simulé, malgré qu'ils assurent de trouver un minimum global, sont gourmands en temps de calculs. Pour palier cet inconvénient, d'autres types d'algorithmes déterministes, dont l'algorithme ICM (*Iterated Conditional Modes*) ont été proposés.

### 3.3.2 Recuit Simulé (RS)

Cet algorithme stochastique repose sur l'idée de l'association d'une procédure de *Monte Carlo* d'échantillonnage (algorithme de *Metropolis* ou échantillonneur de Gibbs) à un schéma de refroidissement en température [48]. Dans notre contexte, le but est de trouver la configuration  $\hat{x}_{MAP}$  qui maximise la probabilité  $P_{X|Y}(x|y)$ .

D'un point de vue pratique, le choix du mode de décroissement de la température vers 0 est très important et peut s'avérer délicat [65]. Les meilleurs résultats sont obtenus avec un mode de convergence assez lent vers 0 et si le nombre d'itérations tends vers l'infini. Ceci permet dans une grande mesure d'atteindre le minimum global de la fonction d'énergie. On peut remarquer que si la température est très élevée, l'énergie peut avoir des accroissements importants, ce qui permet de sortir des faux



minimums. Pour que le RS puisse atteindre le vrai minimum global de la fonction d'énergie, il faut que les conditions de convergence que soit la décroissance de la température ou le nombre fini d'itérations soient respectées.

□ **Algorithme** : Recuit Simulé $T_0$  Température initiale $T_{min}$  Température finale $\hat{x}^{[0]}$  Configuration initiale $F(.)$  Fonction décroissante de la température

- $T \leftarrow T_0$
- Calcul de  $\hat{x}^{[k+1]}$  à partir de  $\hat{x}^{[k]}$ 
  1. On balaie l'ensemble des sites  $s$  (peu importe l'ordre) et :
    - En chaque site  $s$ , on calcule  $P_{X_s|X_{v_s}, Y_s}(x_s|\eta_s, y_s y)$  pour chacune des classes possibles :

$$P_{X_s|X_{v_s}, Y_s}(x_s|\eta_s, y_s y) = \frac{1}{Z_s} \exp \left\{ -\frac{1}{T_k} \left( \sum_{c \in C} V_c(x_s) + V(x_s, y_s) \right) \right\} (\forall x_s \in \Omega)$$

- On tire l'une des classes  $x_s$  selon les probabilités d'appartenance  $P_{X_s|X_{v_s}, Y_s}$  définie précédemment.

$$\hat{x}^{[k+1]} \leftarrow x$$

2. Faire décroître la température  $T^{[k+1]} = F(T^{[k]})$  et incrémenter l'itération  $k = k + 1$ .
3. Retour en 1. jusqu'à la satisfaction d'un critère d'arrêt. Généralement :
  - si  $T_k \geq T_{min}$  retour en 1.

TAB. 3.3 – *Algorithme RS*

### 3.3.3 Iterated Conditional Modes (ICM)

L'ICM [4] a été proposé comme une méthode itérative qui permet d'obtenir une solution approximative de l'estimateur **MAP**. C'est un algorithme déterministe nécessitant une bonne initialisation. L'ICM est une version du RS avec une température nulle ( $T=0$ ). Contrairement au RS qui visite plusieurs puits (minimums) d'énergie puis se stabilise dans l'un d'eux, L'ICM converge vers le premier minimum local rencontré. Il est sensible à l'initialisation de départ et au mode de balayage de l'image. Il peut converger plus rapidement au minimum global lorsqu'il dispose d'une bonne configuration initiale (i.e., assez proche de la configuration optimale).

□ **Algorithme : Iterated Conditional Modes (ICM)**

|  $\hat{x}^{[0]}$  **Configuration initiale**

- Initialisation : on choisit une configuration initial  $\hat{x}^{[0]}$ , aussi proche que possible de la configuration optimale. On peut prendre par exemple une segmentation au sens du **MV** :

$$\hat{x}_s^{[0]} = \arg \max_{x_s} \{P_{Y_s|X_s}(y_s|x_s)\} \quad (\forall s \in S)$$

- Calcul de  $\hat{x}_s^{[k+1]}$  à partir de  $\hat{x}_s^{[k]}$  :

$$x \leftarrow \hat{x}^{[k]}$$

1. On balaie l'ensemble des sites  $s$  (selon une stratégie de visite de sites) et :

- En chaque site  $s$ , on calcule  $P_{x_s|X_{v_s}, Y_s}(x_s|\eta_s, y_s)$  pour chacune des classes possibles

$$P_{x_s|X_{v_s}, Y_s}(x_s|\eta_s, y_s) = \frac{1}{Z_s} \exp\left\{-\left(\sum_{c \in C} V_c(x_s) + V(x_s, y_s)\right)\right\} (\forall x_s \in \Omega)$$

- On sélectionne la classe  $x_s$  qui correspond à la probabilité maximale.

$$\hat{x} = \arg \max_{x_s \in \Omega} \{P_{X_s|Y_{v_s}, Y_s}(x_s|\eta_x, y_s)\}$$

$$\hat{x}^{[k+1]} \leftarrow x$$

2. Retour en 1. jusqu'à la réalisation d'un critère d'arrêt. Généralement :

$$\text{si } \hat{x}_s^{[k+1]} \neq \hat{x}_s^{[k]} \quad k \leftarrow k + 1 \text{ et retour en 1.}$$

TAB. 3.4 – *Algorithme ICM*

## Chapitre 4

# Segmentation d'images par partitionnement de graphes

### 4.1 Introduction

L'algorithme de segmentation proposé dans ce mémoire est hiérarchique avec deux niveaux d'abstraction. Dans un premier temps, celui-ci est basé sur une segmentation Markovienne non-supervisée dont les paramètres sont estimés automatiquement grâce à la procédure stochastique ECI décrite dans le chapitre 3. Ce processus de segmentation s'effectue au niveau du pixel et permet de construire des régions cohérentes au sens des niveaux de gris. Dans cette segmentation Markovienne au sens des niveaux de gris, l'image est en fait modélisée par un graphe où les différents pixels sont les nœuds de ce graphe et les nœuds sont connectés à leurs 4 plus proches pixels voisins.

Dans un second temps, une segmentation au sens de la texture consistera à fusionner ensemble certaines régions (obtenues précédemment). Dans ce but, une analyse de graphe permettra de réduire le nombre de régions et d'avoir des régions plus grandes, disjointes qui représentent les objets discernables de l'image à segmenter. Cette fusion de régions sera fondée sur la *ressemblance texturale* de ces régions.

Plus précisément comme nous l'avons dit dans le paragraphe précédent, on doit tout d'abord segmenter l'image en niveaux de gris. Ce processus sera effectué par une segmentation Markovienne qui générera une première carte de régions que nous appellerons  $R$ , homogène au sens des niveaux de gris. Cette carte de segmentation est *sur-segmentée*, c'est-à-dire qu'elle présente de nombreuses régions parfois non significatives (les différents objets de la scène sont en fait divisés en plusieurs zones homogènes au sens des niveaux de gris). Dans le but d'avoir une segmentation au sens de la texture, où les régions de segmentation représenteraient réellement des objets élémentaires connexes (parties significatives de l'images) qui composent l'image originale, une procédure de fusion de régions est ensuite proposée.

Les algorithmes de regroupement de régions adjacentes doivent utiliser des critères de regroupement (de régions) qui sont fondés sur des distances de similarité entre régions pour pouvoir distinguer entre les différentes textures de l'image. Avant de procéder à la fusion, on privilégie tout d'abord l'absorption des petites régions (i.e., celles dont la taille (notée  $\Gamma$ ) est inférieure à la taille du plus petit objet structurant que l'on désire discerner dans l'image) par les grandes régions. Cette opération d'absorption est contrôlée par une simple technique de filtrage de telle sorte que chaque petite région sera absorbée par la région avec laquelle celle-ci est la plus connectée.

Nous utiliserons deux méthodes pour le regroupement des régions. La première méthode (chapitre 5) consiste à utiliser le partitionnement par les vecteurs propres [71]. La deuxième méthode est la méthode de flux ou Markov Clustering (MCL) développée par *Stijn Van Dangen* dans le cadre du partitionnement de graphes de grandes tailles [19]. Cette méthode sera présentée au chapitre ??.

Dans ces deux méthodes, un graphe de régions est modélisé par un graphe d'adjacence de régions, associé à une matrice d'adjacence qui sera ensuite analysée. Analyser un graphe d'adjacence de régions revient à comparer ces régions adjacentes deux à deux pour décider d'un éventuel regroupement en une seule. A cet effet nous évaluons le degré de similarité entre elles. La décision est fondée sur la comparaison des degrés de similarité et sera mesurée grâce à des mesures de ressemblance entre imageries qui

seront décrites dans la section 4.4.2.

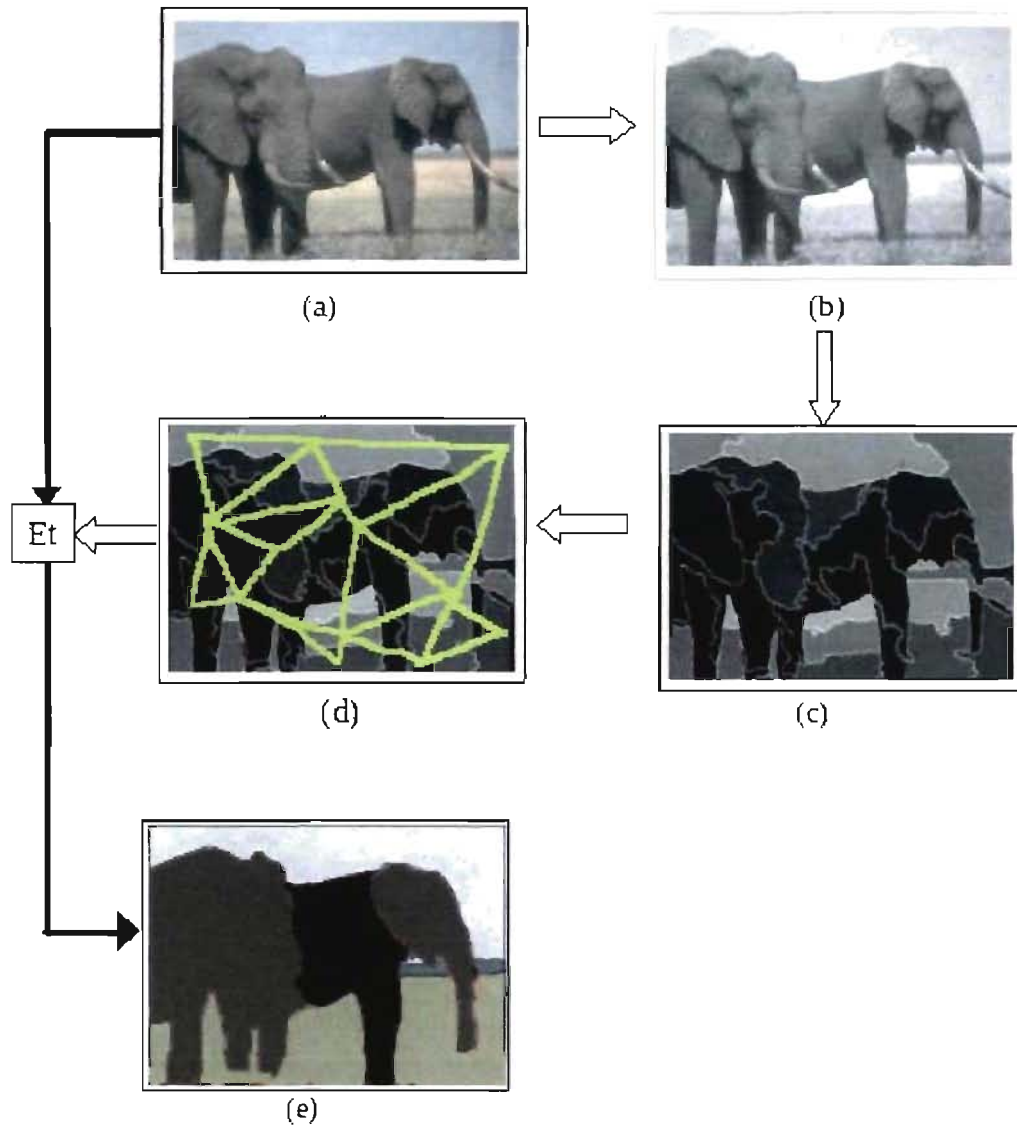


FIG. 4.1 – Diagramme général de l'algorithme proposé : (a) image en couleur, (b) image en niveaux de gris, (c) segmentation au sens des niveaux de gris (ICM, 29 régions), (d) carte et graphe de régions, (e) fusion de régions au sens de la texture (6 régions).

## 4.2 Notions de base sur les graphes

Un graphe est un objet mathématique constitué d'un ensemble  $V$  de noeuds et d'un ensemble  $E$  d'arcs reliant certain de ces noeuds. Les graphes sont donc particulièrement adaptés à la représentation de relations entre des éléments d'une image (i.e., les régions en niveaux de gris, en couleurs ou en textures d'une image).

Un graphe  $G$  peut être écrit par la relation  $G = \{V, E\}$  dans laquelle chaque arête peut être représentée par une paire de noeuds. Un graphe est souvent dessiné comme un ensemble de points connectés par des arcs (cf. figure 4.2).

- **Graphe orienté** : C'est un graphe dans lequel les arcs  $(a, b)$  et  $(b, a)$  sont distincts. Dans ce type de graphe, les arcs sont des flèches indiquant leur orientation.
- **Graphe non orienté** : C'est un graphe dans lequel il n'y a pas de distinction entre les arcs  $(a, b)$  et  $(b, a)$ .
- **Graphe pondéré** : C'est un graphe dans lequel des poids sont associés aux arcs.
- Deux noeuds sont appelés **connectés**, s'il existe une séquence d'arcs, dont l'un de ces deux noeuds est l'extrémité initiale et l'autre est l'extrémité finale de cette séquence.
- **Graphe connecté** : C'est un graphe où toutes les paires de noeuds sont connectés.
- Chaque graphe se compose d'un ensemble de composants connectés disjoints  $G = \{V_1, V_2, \dots, V_n; E_1, E_2, \dots, E_n\}$ , où  $\{V_i, E_i\}$  sont tous les {sous-graphes connectés}.

## 4.3 Modélisation d'images par un graphe

Une image est une population d'entités organisées selon une certaine géométrie. Il est particulièrement intéressant de modéliser cette population à l'aide d'un graphe.



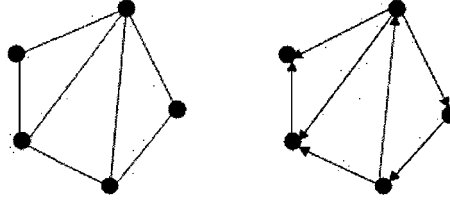


FIG. 4.2 – Graphe non-orienté (à gauche) et graphe orienté (à droite)

Dans [62] *Shi et Malik* considèrent la segmentation d'images comme un processus de regroupement de ces entités. Ce regroupement lui-même est considéré comme un problème de partitionnement d'un graphe où les noeuds (entités) peuvent être des pixels, des segments, des contours, des régions, etc., et les arcs pondérés correspondent aux degrés de similarité en intensité, couleur ou texture de ces différentes entités.

Il est utile de comprendre qu'un graphe pondéré peut être aussi représenté par une matrice carrée (cf. figure 4.3). Dans cette matrice il y a une rangée  $i$  et une colonne  $j$  pour chaque noeud. Le  $(i, j)^{ième}$  élément de la matrice représente le poids de l'arc entre le noeud  $i$  et le noeud  $j$ .

## 4.4 Construction de graphe de régions

Comme nous l'avons déjà dit, l'image est segmentée au sens des niveaux de gris, en premier lieu, par une approche Markovienne. L'image ainsi obtenue est *sur-segmentée* (i.e., sur-divisée) et ne définit pas toujours les frontières réelles qui séparent les objets élémentaires de l'image originale. Dans le but d'avoir une carte de segmentation qui représente ces différentes entités texturales de la scène, l'image *sur-segmentée* est convertie en une carte de régions  $R$ . Cette dernière est ensuite modélisée par un graphe d'adjacence de régions dans lequel chaque noeud représente une région et chaque arc est représenté par un poids qui mesurera une mesure de similarité texturale entre deux régions. Nous pouvons nous servir des termes utilisés dans la thèse de *P. Colantoni*

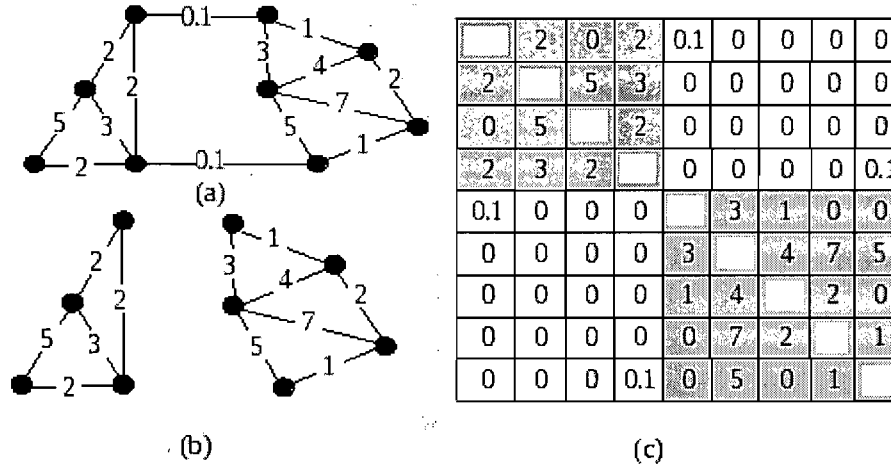


FIG. 4.3 – (a) Graphe non-orienté pondéré, (b) Graphe partitionné en deux sous-graphes disjoints, (c) Matrice d'affinité bloc-diagonalisée

[13] pour associer une valuation à chaque noeud et une pondération à chaque arc. Dans notre cas de segmentation d'images couleurs, la valuation est la couleur moyenne des pixels d'une région, et la pondération se calculera par la distance entre les couleurs moyennes des régions adjacentes. Le choix de la distance sera détaillé en section 4.4.2.

Pour caractériser la ressemblance entre les différentes régions connexes (de la carte de régions  $R$ ), on estimera pour chaque région une distance qui mesurera la ressemblance de sa texture avec celles de ses voisines. Plus précisément ce calcul sera fait pour chaque imagerie (de taille  $5 \times 5$  par exemple) à l'intérieur de chaque région. Afin de rendre plus robuste ce calcul et prendre en compte la diversité texturale qui existe au sein d'une même texture, on retiendra les huit mesures les plus similaires (i.e., les plus petites). Par sommation des distances calculées entre ces (huit) mesures calculées sur ces huit paires d'images, on en déduira les relations en termes de distances entre les régions. Cette interaction inter-régions est modélisée mathématiquement par un graphe d'adjacence de régions (GAR) où les nœuds sont des régions et les arcs sont les distances entre régions. Lorsque les distances entre plusieurs régions sont relativement

petites ou nulles, elles peuvent être fusionnée afin de ne former qu'une seule région cohérente au sens de la texture. Cette fusion est réalisée grâce à l'analyse de la théorie des graphes qui permet le partitionnement de graphe en plusieurs sous-graphes.

Algorithmiquement, soit  $R$  la carte de régions obtenue par la *sur-segmentation* ICM, de longueur  $L$  et largeur  $W$  et comportant un nombre de régions noté  $N_r$ . Soit une imagerie  $F$  de taille  $M^2$  ( $M$  est un nombre entier impair inférieur à  $L$  et à  $W$ ) appelée *fenêtre de recherche*. (cf. figure 4.4 (c,d,e)). La fenêtre de recherche  $F$  est centrée sur une petite imagerie  $\check{F}_p$ , de taille  $N^2$  ( $N$  est un nombre entier impair inférieur à  $M$ ) et centrée sur le pixel  $p$ . On réalise un déplacement à recouvrement dans l'imagerie  $F$ , par une fenêtre  $\check{G}_q$  de même taille que  $\check{F}_p$ , centrée sur le pixel  $q$ . Les distances de similarité entre l'imagerie  $\check{F}_p$  et chaque imagerie  $\check{G}_q$  sont ensuite calculées pour tout pixels  $p$  et  $q$  n'appartenant pas à la même région mais à deux régions voisines.

Une fois que toutes les distances entre l'imagerie  $\check{F}_p$  et toutes les images  $\check{G}_q$  nécessaires pour le déplacement complet de la grande imagerie  $F$  sont calculées puis triées en ordre croissant, on retient que les  $Z$  ( $Z = 8$  dans la pratique) premières distances les plus petites (correspondant aux images  $\check{G}_q$  les plus similaires à l'imagerie  $\check{F}_p$ ).

Par l'itération du même processus sur tout l'espace de la carte de régions  $R$ , on peut aussi calculer les distances totales entre les différentes régions deux à deux.

À partir des données précédentes (distances entre les images), on peut calculer les distances pondérées (voir section 4.4.2) entre les régions et construire notre graphe ainsi que la matrice d'adjacence associée.

#### 4.4.1 Distance de similarité entre deux régions

La distance entre deux images  $\check{F}_p$  et  $\check{G}_q$  est calculée à partir des composantes colorimétriques des pixels contenues dans ces deux images. Le choix de l'espace

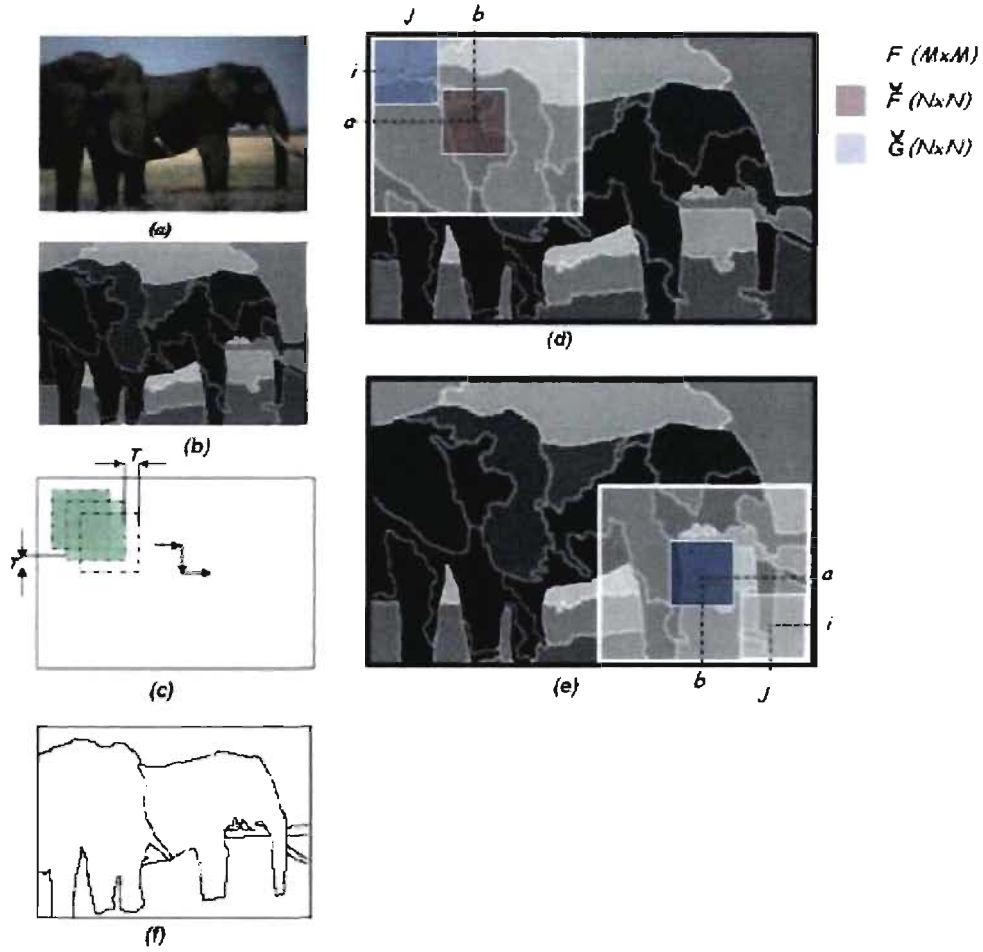


FIG. 4.4 – (a) Image originale. (b) carte de régions  $R$ . (c),(d),(e) parcours de la carte  $R$  de gauche à droite, de haut en bas ( $M = 15$ ,  $N = 5$ ). (f) régions fusionnées (segmentation voulue).

de la couleur dans la segmentation d'images couleur reste toujours un défi. En effet, il existe plusieurs espaces de couleurs et aucun d'entre eux n'est optimal pour le problème de la segmentation. Les espaces de couleurs les plus utilisés sont  $RVB$  et  $YUV$  mais ces espaces ne sont pas uniformes du point de vue perceptuel (i.e., deux couleurs qui sont proches dans l'espace des couleurs [au sens de la distance euclidienne

le plus souvent], peuvent paraître assez différents pour l'oeil). Par contre dans l'espace  $LAB$ , deux couleurs proches en distance le sont aussi pour l'oeil. Un autre avantage de cette représentation est qu'elle permet une séparation de l'information couleur en un canal de luminosité  $L^*$  et deux canaux chromatiques  $a$  et  $b$ . Pour plus de détail sur les couleurs, voir [21] et [56].

Dans notre cas, nous avons pensé combiner l'utilisation de ces six espaces de couleurs ;  $RVB$ ,  $HSV$ ,  $LUV$ ,  $YIQ$ ,  $XYZ$  et  $L^*ab$  pour bénéficier de leurs avantages de représentation et d'utiliser le maximum d'informations possibles.

#### 4.4.2 Choix de la distance

Le choix de la métrique pour mesurer la similarité texturale entre imagerie est important. Afin de mieux percevoir les couleurs, plusieurs techniques de calcul de distance ont été proposées par *Helmholtz (1896)*, *Shrodinger (1920)*, *Stiles (1946)* et plus récemment par *Vos and Walraven (1972)*. Des études de colorimétrie ont montré que pour tenir compte de la réalité visuelle on devait munir l'espace des couleurs d'une distance Riemannienne. Cette métrique Riemannienne est approchée par la distance Euclidienne [20][18][72].

Dans notre cas, la distance  $d(\check{F}_p, \check{G}_q)$  entre deux imagerie  $\check{F}_p$  et  $\check{G}_q$  est définie par la relation

$$d(\check{F}_p, \check{G}_q) = \exp \frac{-\sum_{i=1}^{N_c} (\mu_{pi} - \mu_{qi})^2}{\sigma} \quad (4.1)$$

Avec  $\mu_{pi}$  ( $\mu_{qi}$ ) la valeur moyenne de la composantes colorimétrique  $c_i$  associée à l'imagerie  $\check{F}_p$  (respectivement  $\check{G}_q$ ),  $N_c$  est le nombre de composantes colorimétriques utilisées et  $\sigma$  est un paramètre d'échelle qui permettra de différencier peu ou beaucoup chacune de ces différentes distances. Nous prendrons 6 espaces couleurs ( $RVB$ ,  $HSV$ ,  $XYZ$ ,  $LUV$ ,  $YIQ$  et  $LAB$ ) et donc  $N_c = 3 \times 6 = 18$  composantes colorimétriques associées à la valeur moyenne respectivement de la composante R, V, B, H, S, V, X,

Y, Z, L, U, V, Y, I, Q, L, A, B.

Finalement la distance  $d_r$  entre deux régions quelconques  $r_i$  et  $r_j$  est ensuite donnée par l'équation suivante

$$\mathcal{D}_r(r_i, r_j) = \frac{1}{|r_i|} \sum_{p \in r_i} \sum_{q \in r_j} \Psi[d(\check{F}_p, \check{G}_q)] \quad (4.2)$$

Où  $|\cdot|$  est le cardinal d'une région (ici on divise par  $|r_i|$  pour favoriser la fusion des petites régions  $r_i$  par une région plus grande  $r_j$ ),  $p$  et  $q$  sont des pixels,  $\Psi[\cdot]$  l'opérateur qui calculera la moyenne des huit distances  $d(\check{F}_p, \check{G}_q)$  les plus petites. Le graphe déduit par le calcul de toutes les distances entre les différentes régions est un graphe orienté noté  $\dot{G}_r$  ( $\dot{G}_r(r_i, r_j) \neq \dot{G}_r(r_j, r_i)$ ). Le graphe non orienté noté ( $G_r$ ) que l'on peut considérer dans la suite sera défini par :

$$G_r(r_i, r_j) = \max(\dot{G}_r(r_i, r_j), \dot{G}_r(r_j, r_i)) \quad (4.3)$$

ou l'équation :

$$G_r(r_i, r_j) = \frac{1}{2}(\dot{G}_r(r_i, r_j) + \dot{G}_r(r_j, r_i)) \quad (4.4)$$

Les expériences montrent que l'équation 4.3 permet d'augmenter la qualité de la segmentation par rapport à l'équation 4.4. L'interprétation derrière ça, est que nous nous intéressons à favoriser la liaison maximale entre deux régions peu importe la direction entre ces deux régions, et pas la liaison moyenne qui peut s'affaiblir si l'une des direction a un faible poids (au voisinage de zéro par exemple). Prenons l'exemple suivant :  $\dot{G}_r(r_1, r_2) = 0.9$  et  $\dot{G}_r(r_2, r_1) = 0.2$ , Alors, on préfère prendre  $G_r(r_1, r_2) = 0.9$  (Eq. (4.3)) plutôt que de prendre  $G_r(r_1, r_2) = 0.55$  (Eq.(4.4)).

## Chapitre 5

# Méthodes de partitionnement du graphe de régions

### 5.1 Introduction

Nous allons décrire les méthodes permettant de fusionner les  $K$  régions  $r_i$  ( $i = 1, \dots, K$ ) de la carte de *sur-segmentation*  $R$  (définie dans le chapitre 4) afin d'obtenir une autre carte  $R'$  de  $L$  régions  $r'_j$  ( $j = 1, \dots, L$  et  $K > L$ ). Chacune de ces  $L$  régions est appelée un *cluster*, cette nomination est utilisée pour rester dans le contexte du partitionnement de graphe de petits *clusters* en un nombre réduit de grands *clusters*. La segmentation par coupe de graphe est une approche intéressante et décrite en [71][62][63] qui postule que

- Chaque région  $r_i$  est considérée comme un nœud dans le graphe.
- La similarité entre deux régions est représentée par le poids de l'arc reliant ces deux régions.
- La segmentation est réalisée en coupant certains arcs dans le graphe pour former des clusters disjoints.

Les poids intra-clusters (forts) sont fortement comparables par rapport aux poids inter-clusters (quasiment nuls). Comme résultat, le problème de la segmentation d'images devient un problème du partitionnement de graphe en clusters. Mathématiquement, nous pouvons représenter une image (dans notre cas une carte de  $K$  régions) par une matrice  $K \times K$ , appelée matrice de similarité ou matrice d'affinité dont les coefficients  $A_{ij}$  sont donnés par :

$$A_{ij} = d_r(r_i, r_j) \quad (5.1)$$

Où  $d_r(r_i, r_j)$  est définie par l'équation (4.2).  $A_{ij}$  indique la similarité entre les deux régions  $r_i$  et  $r_j$ . Étant donné une telle matrice, le problème de la segmentation se réduit à savoir si on peut bloc-diagonaliser cette matrice d'affinité par l'utilisation de la méthode des vecteurs propres ou si on peut décomposer le graphe original en sous-graphes disjoints par la méthode de flux (MCL).

## 5.2 Bloc-diagonalisation de la matrice d'affinité par les vecteurs propres

Comme solution [71], si nous changeons les rangs et les colonnes de la matrice  $A$  (changer l'ordre) nous pouvons former des blocs (*clusters*) dans la matrice d'affinité (cf. figure 5.1). Nous introduisons une nouvelles variable de poids pour chaque *cluster* de composantes  $\omega_n$ ,

$$\omega_n = \begin{bmatrix} \omega_{n1} \\ \omega_{n2} \\ \vdots \\ \omega_{nK} \end{bmatrix} \quad (5.2)$$

Où  $\omega_{ni}$  signifie le poids de la  $i^{\text{ème}}$  région associée au  $n^{\text{ème}}$  cluster. C'est-à-dire,



plus le poids est fort, plus l'association est forte. Habituellement, un tel vecteur de poids est normalisé, c'est-à-dire que nous avons :

$$\omega_n^T \omega_n = 1 \quad (5.3)$$

En utilisant cette indication de variable de poids, le problème de bloc-diagonalisation peut être formulé par un problème de programmation linéaire, c'est-à-dire :

$$\omega_n^* = \max_{\omega_n} \omega_n^T A \omega_n, \quad \text{sujet à } \omega_n^T \omega_n = 1 \quad (5.4)$$

Pour résoudre ce problème d'optimisation, nous utilisons le Lagrangian

$$\omega_n^T A \omega_n + \lambda(\omega_n^T \omega_n - 1) \quad (5.5)$$

Par différenciation et en laissant tomber les facteurs des deux cotés de l'équation, Nous aurons :

$$A \omega_n^* = \lambda \omega_n^* \quad (5.6)$$

Ce qui signifie que la solution  $\omega_n^*$  est un vecteur propre de la matrice d'affinité  $A$ . Donc, cette approche propose une solution très intéressante à notre problème.

Numériquement cette méthode consiste à effectuer la décomposition de la matrice  $A$  en valeurs propres, puis identifier les  $C$  plus grandes valeurs propres  $\lambda_c$ ,  $c = 1, \dots, C$ , et trouver les vecteurs propres correspondants  $e_c$ ,  $c = 1, \dots, C$ . Ici,  $e_c$  révèle l'identité de chaque région pour le  $c^{ième}$  cluster, puisque si la valeur  $e_{c_i}$  tends vers zéro, la  $i^{ième}$  région n'appartient pas au  $c^{ième}$  cluster, tandis que si cette valeur est grande (supérieure à un seuil), la région appartient à ce cluster.

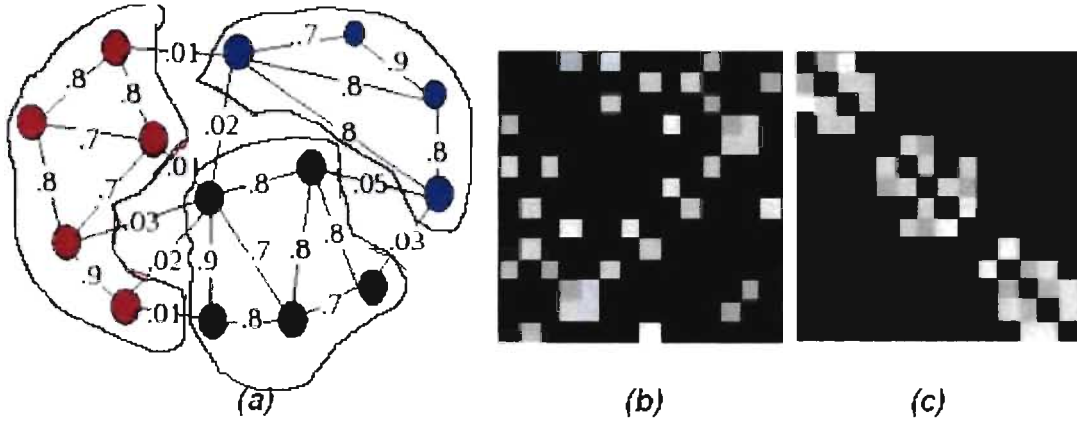


FIG. 5.1 – (a) Ensemble de données. (b) matrice de similarité, (c) matrice de similarité après réorganisation des lignes et des colonnes de la matrice (b), la matrice (c) est bloc-diagonale composée de trois blocs apparents.

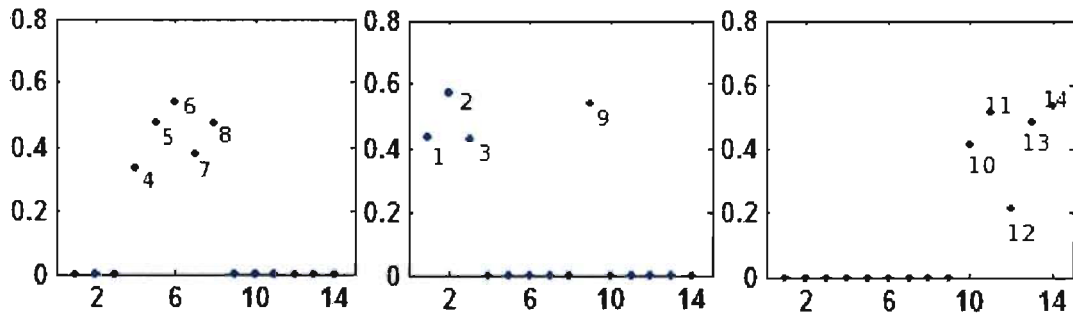


FIG. 5.2 – Les trois vecteurs propres correspondant aux trois plus grandes valeurs propres (les axes des abscisses correspondent aux données (numéros de régions) et les axes des ordonnées correspondent aux valeurs des vecteurs propres) de la matrice de similarité de l'ensemble des données de la figure 5.1. Les régions 4, 5, 6, 7 et 8 définies par le 1<sup>er</sup> vecteur propre appartiennent au 1<sup>er</sup> cluster, les régions 1, 2, 3 et 9 définies par le 2<sup>ème</sup> vecteur propre appartiennent au 2<sup>ème</sup> cluster et les régions 10, 11, 12, 13 et 14 définies par le 3<sup>ème</sup> vecteur propre appartiennent au 3<sup>ème</sup> cluster.

### 5.3 Segmentation par la méthode de flux (MCL)

Une autre méthode pour le regroupement des régions d'une carte *sur-segmentée* est décrite par l'algorithme MCL (Markov Clustering) développé par *Slijm Van Dongen*

□ **Algorithme** : Segmentation par les vecteurs propres

**Entrée** :  $M$  Matrice de similarité des régions, seuil  $s$

1. Calculer les vecteurs et les valeurs propres de la matrice  $M$ .

2. *Répéter*

- Prendre le vecteur propre correspondant à la plus grande valeur propre qui n'est pas encore traitée ; mettre à zéro toutes les composantes correspondantes aux éléments qui ont été déjà partitionnés, et seuiller les composantes restantes pour déterminer quel élément appartient au cluster en cours. Le seuil  $s$  est choisi d'avance.

- Si tous les éléments ont tous comptés, il y a un nombre suffisant de clusters

*Jusqu'à* Avoir un nombre suffisant de groupes.

TAB. 5.1 *Algorithme de la segmentation par les vecteurs propres*

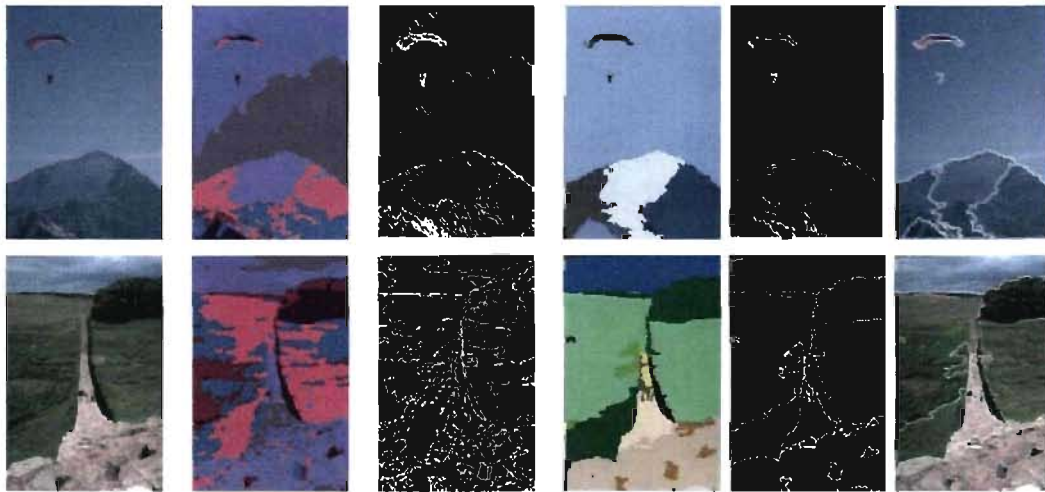


FIG. 5.3 – Segmentation par la méthode des vecteurs propres EIGEN : de gauche à droite, image originale, ICM, contours ICM, EIGEN ( $s=0.4$  en haut et  $s=0.8$  en bas), contours EIGEN, image originale avec contours EIGEN.

[19] dans le cadre du partitionnement de graphes de grandes tailles. La technique de partitionnement par le MCL repose sur une idée simple : un parcours aléatoire d'une partie dense d'un graphe a peu de risque de quitter cette partie dense avant d'avoir visité un bon nombre de sommets. Plutôt que de simuler des marches aléatoires, l'algorithme propose d'étudier le flux du graphe en se basant sur un processus de Markov. La matrice de transition est successivement élevée à la puissance  $e$  (simulant  $e$  marches) puis normalisée. L'algorithme proposé converge vers un point fixe ou vers un état récurrent. Les composantes connexes du graphe induit par la matrice finale sont les groupement de la partition (cf. figure 5.4).

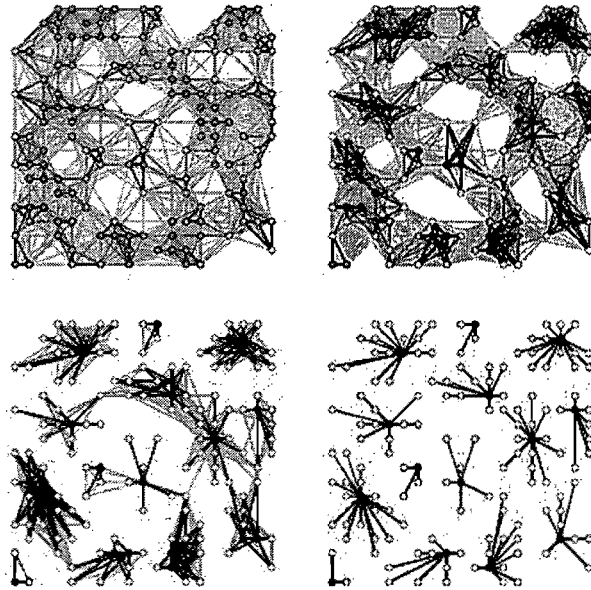


FIG. 5.4 – Partitionnement successif du graphe par MCL (source [19])

Soit  $M$  la matrice d'adjacence du graphe  $G = (V, E)$ ,  $e$  le facteur d'expansion et  $r$  le facteur d'inflation. Les différentes étapes de l'algorithme MCL sont décrites ci-dessous :

- $M$  est élevée à la puissance  $e$  (généralement fixé à 2).
- Chaque élément de la matrice (poids) est élevé à la puissance  $r$  (à régler).

- Chaque poids est ensuite divisé par le poids total de la ligne (normalisation).
- L'algorithme est réitéré tant qu'aucun point fixe ou état récurrent n'ait été atteint.

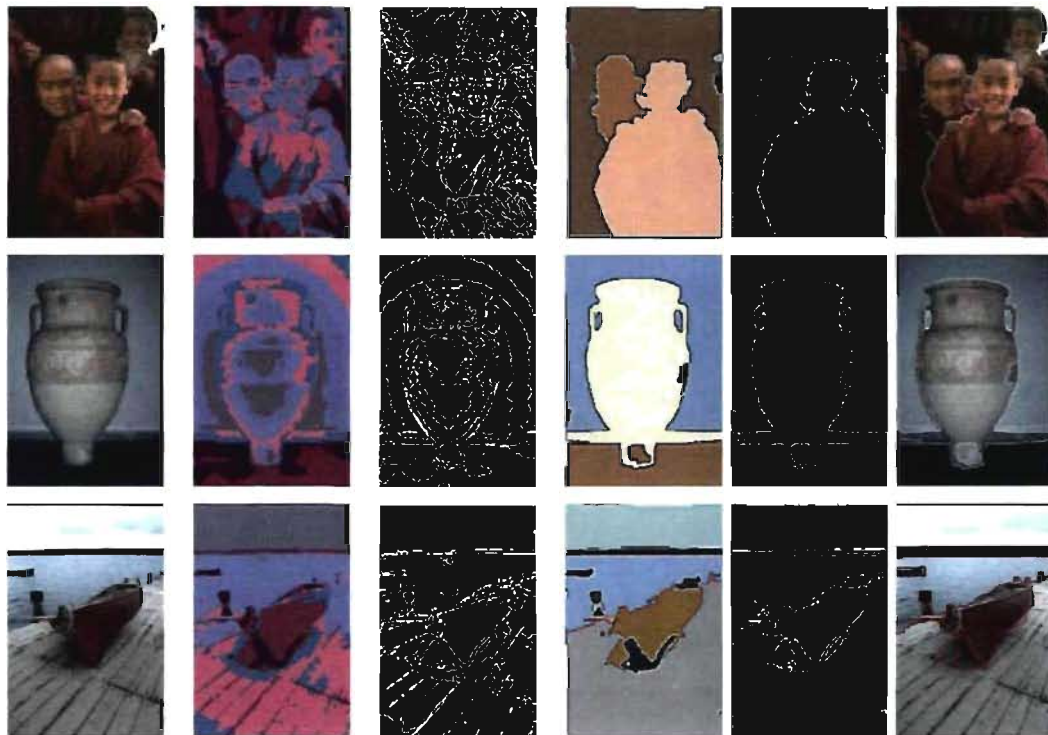


FIG. 5.5 – Segmentation par MCL : de gauche à droite, image originale, ICM, contours ICM, MCL (en haut  $r=1.1$ , au milieu  $r=1.2$  et  $r=1.26$  en bas), contours MCL, image originale avec contours MCL.

□ **Algorithme** : Markov Clustering (MCL)

**Entré** : Graphe de régions  $G = (V, E)$ , paramètre d'expansion  $e$ , paramètre d'inflation  $r$ .

1.  $M \leftarrow M(G)$ . Matrice d'affinité associé à  $G$
2. **TantQue**  $M$  n'est pas un point fixe
  - $M \leftarrow M^e$
  - PourTout**  $u \in V$ 
    - PourTout**  $v \in V$ 
      - $M_{uv} \leftarrow M_{uv}^r$
    - fin**
  - PourTout**  $v \in V$ 
    - $M_{uv} \leftarrow \frac{M_{uv}}{\sum_{w \in V} M_{uw}}$
  - fin**
- fin**
- FinTantQue**
3.  $H \leftarrow$  graphe induit par les entrées non-nules de  $M$
4.  $C \leftarrow$  groupes induits par les composantes connexes de  $H$

TAB. 5.2 – *Algorithme de segmentation par MCL*

## Chapitre 6

### Re-segmentation

Il est parfois difficile d'éviter le phénomène de la *sur-segmentation* (i.e., trop de régions dans la segmentation finale) en utilisant les méthodes de partitionnement de graphes contrôlées par des paramètres d'ajustement. En effet il n'est pas toujours possible de trouver les paramètres optimaux quand ces derniers existent.

Dans nos deux méthodes de regroupement des régions par partitionnement de graphes, la précision de la segmentation de l'image est reliée à quelques paramètres ajustables, le premier est le seuil  $s$  utilisé dans la méthode des vecteurs propres et le deuxième est le paramètre d'inflation  $r$  utilisé dans la méthode MCL.

Par exemple sur la figure 6.1, quand  $r = 1.5$ , on obtient un modèle presque exact de la carte de régions initiale, comportant une centaine de détails (figure 6.1 (c)), et à l'inverse, pour une valeur plus petite de  $r$ , l'image segmentée se résumera à quelques régions seulement (figure 6.1 (d)).

Quand on ne dispose pas d'un *à priori* sur le nombre ou la taille probable des différents objets à délimiter, nous proposons une autre façon de réduire le nombre de régions à considérer. A cet effet, une partition initiale en régions est tout d'abord calculée (carte  $R$  de *sur-segmentation* obtenue par ICM) puis son graphe d'adjacence (GAR) est construit, et une segmentation est obtenue en regroupant les régions de

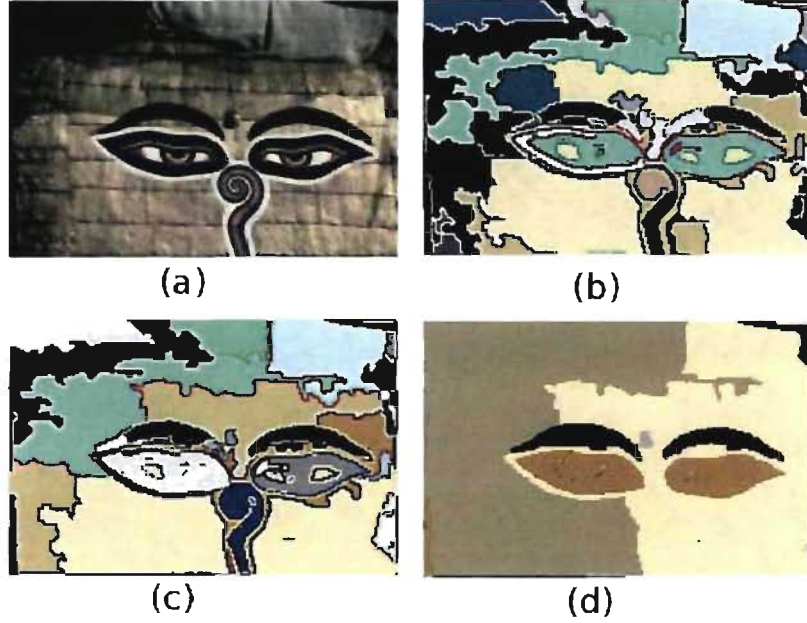


FIG. 6.1 – Segmentation MCL : (a) image originale, (b) régions ICM, (c)  $MCL_{\tau=1.5}$ , (d)  $MCL_{\tau=1.25}$

la carte  $R$  par l'une des deux méthodes discutées précédemment (EIGEN ou MCL) dans un deuxième temps. Ce processus est ainsi réitéré plusieurs fois, jusqu'à la réalisation d'un critère d'arrêt. Le processus d'itérations (i.e., *re-segmentation*), est réalisé en faisant décroître lentement le paramètre de régularisation  $\tau$  (dans le cas de *re-segmentation* par MCL, voir figure 6.2) ou accroître le seuil  $s$  (dans le cas de *re-segmentation* par les vecteurs propres) de telle façon que d'une itération à l'autre il n'y ait que les régions les plus marquantes qui se détachent de l'image. La stratégie établie pour la décroissance (la croissance) du paramètre de régularisation  $\tau$  (respectivement du paramètre  $s$ ) est décrite par les équations suivantes :

$$\tau = \max \left\{ \beta_r \exp \left[ \frac{-nbiter}{\tau_r} \right], 1 \right\} \quad (6.1)$$



$$s = \beta_s \left( \exp \left[ \frac{nbiter + 1}{\tau_s} \right] - 1 \right) \quad (6.2)$$

Où  $nbiter$  est le numéro d'itération (débutant par 0) et  $\beta_r$ ,  $\tau_r$ ,  $\beta_s$  et  $\tau_s$  sont des constantes positives.

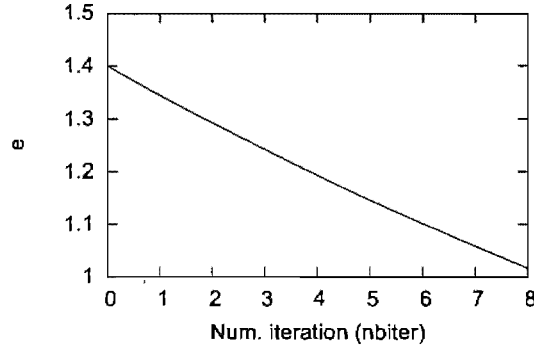


FIG. 6.2 – Évolution du paramètre de régularisation  $r$ ,  $\beta_r = 1.4$ ,  $\tau_r = 25$

## 6.1 A priori sur le nombre de régions dans la segmentation finale

Théoriquement le nombre d'itérations dépend principalement du nombre de régions produites par l'opération de *re-segmentation* Markovienne et de la valeur d'arrivée du paramètre de régularisation  $r$  qui ne doit pas atteindre 1. Ce nombre final de régions (i.e., noté  $N_r$ ) sera un *à priori* pour notre processus de segmentation. Ainsi l'algorithme dépend de trois conditions pour mettre fin à son exécution, la première est le nombre de régions dans la segmentation finale, la deuxième est la valeur finale de  $r$ , et la troisième est le nombre maximal d'itérations (i.e., noté  $nbiter_{max}$ ) de l'algorithme (afin de ne pas *sous-segmenter* l'image, voir figure 6.3 (d) en haut). De ce fait on doit donc procéder de la manière suivante : réitérer le processus de segmentation en

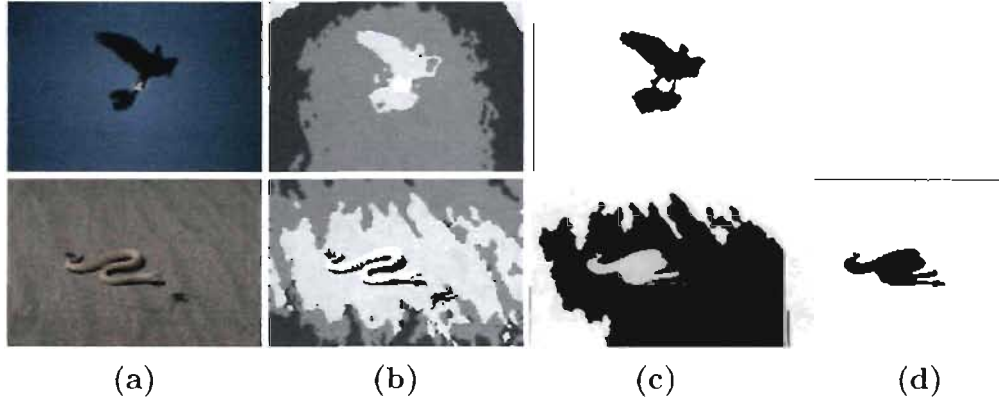


FIG. 6.3 – Effet du nombre d'itérations sur la segmentation finale : (a) image originale, (b) ICM, (c)  $MCL_{nbiter=0}$  (1 itération), (d)  $MCL_{nbiter=1}$  (2 itérations).

faisant décroître  $r$ , vérifier le nombre de régions produites et s'il est encore supérieur à  $N_r$  on continue à réitérer, sinon on arrête l'algorithme. Pratiquement l'algorithme fini par la troisième condition pour plus de 90% des images de la base d'images de Berkeley (BSDb). Par exemple sur la figure 6.3, une itération suffira pour segmenter la première image, tandis que pour la deuxième image, 2 itérations sont nécessaires.

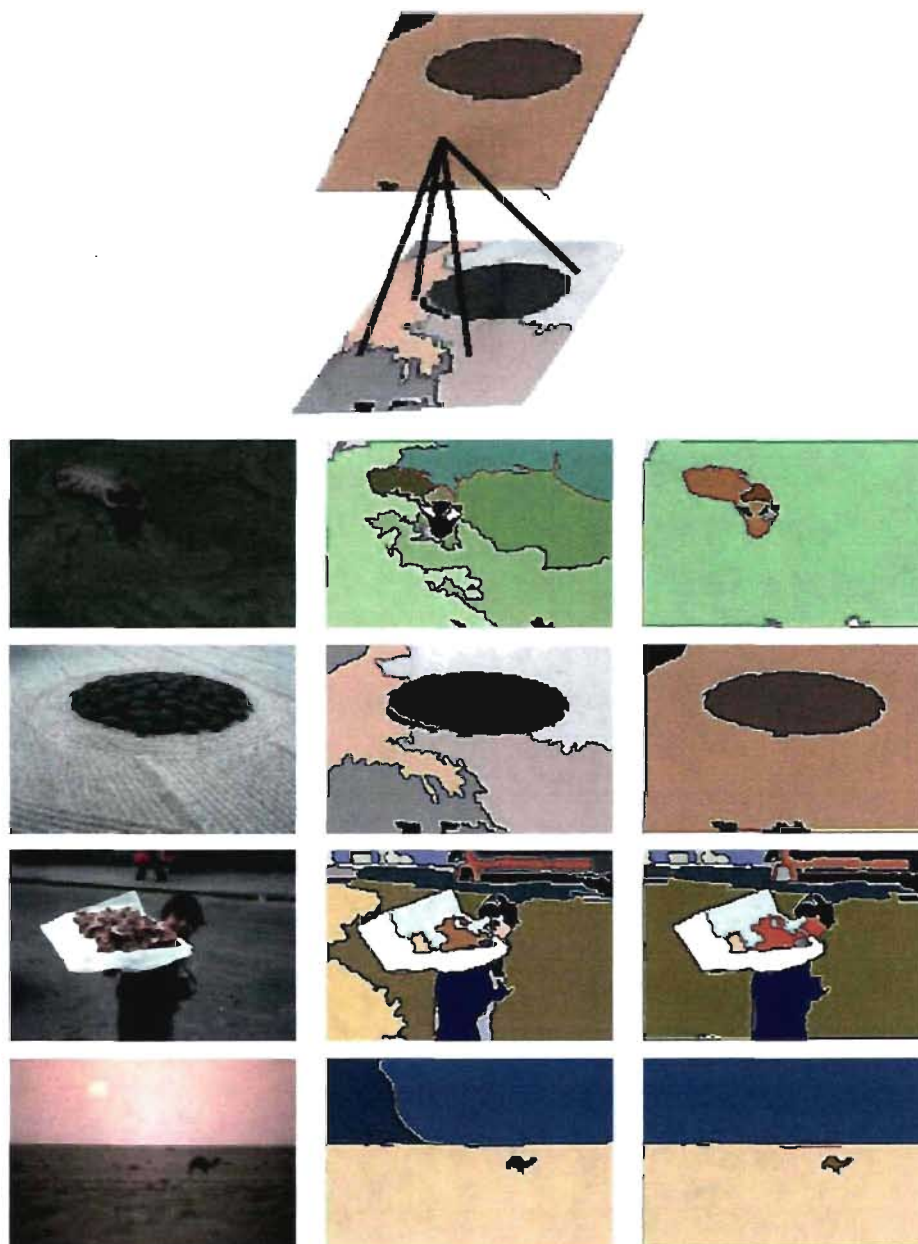


FIG. 6.4 – Re-segmentation MCL : de gauche à droite, image originale, première itération de segmentation ( $r = 1.5$ ), deuxième itération de segmentation ( $r = 1.3$ )

# Chapitre 7

## Evaluation de la segmentation

### 7.1 Introduction

La multiplicité des méthodes de segmentation d'images nécessite d'avoir un critère d'évaluation fiable qui permettra de mesurer le degré de confiance dans le résultat fourni par une méthode de segmentation. Disposer d'un critère d'évaluation de résultats est nécessaire [10][53] :

- aux chercheurs pour comparer un nouvel algorithme à ceux existant, voir même de définir d'autres algorithmes de segmentation par optimisation de critères.
- aux utilisateurs pour choisir un algorithme et régler ses paramètres en fonction du problème à résoudre.

Cependant, il n'existe pas vraiment un critère absolu et meilleur que les autres. Cependant on peut utiliser le maximum de techniques d'évaluation à notre disposition et combiner leurs résultats [44].

De nombreux chercheurs se sont intéressé à ce problème [53][75], et ont proposé plusieurs critères d'évaluation qui sont regroupés en deux classes principales selon que l'on dispose ou non d'une segmentation de référence (*vérité-terrain*) : Critères d'évaluation non-supervisée et critères d'évaluation supervisée. Nous allons nous orienter

dans ce mémoire, vers l'évaluation supervisée.

## 7.2 Évaluation non-supervisée

L'évaluation non-supervisée des résultats de segmentation, consiste à effectuer des mesures d'homogénéité des régions ou de contrastes entre régions, *Stella et Shi* dans [66], ont classifié ces critères en deux grandes catégories : Les critères de Contraste [36][12][11] et les critères d'adéquation au modèle [36][37][5]. Les premiers s'intéressent à la variabilité inter-régions, alors que les seconds s'intéressent à l'uniformité en intensité ou en couleur à l'intérieur des régions.

## 7.3 Évaluation supervisée

L'évaluation supervisée consiste à comparer le résultat fourni par un algorithme avec une segmentation de référence (*vérité-terrain*). Cette dernière peut être une image de synthèse ou une image construite à la main par un expert du domaine de l'application dans le cas d'images naturelles, en traçant des contours (par des médecins, des géographes, etc.), en utilisant des outils informatiques de dessin. Malheureusement, l'évaluation sur images naturelles n'est pas toujours facile à effectuer, parce qu'il n'existe pas de solution unique à la division d'une image en régions adéquates, autrement-dit, la division de l'image en régions est une notion qui dépend absolument de l'application. On peut cependant considérer que la différence entre deux divisions humaines d'une même image est une question de différence de niveau de détails, c'est-à-dire qu'une division peut être une sur-segmentation de l'autre et inversement [7]. Deux manières sont envisageables pour l'évaluation d'une segmentation machine par rapport à une segmentation de référence, la première manière consiste à utiliser les régions dont on trouve les critères GCE (*Global Consistency Error*) [42], PRI (*Probabilistic Rand Index*) [51], et VoI (*Variation of Information*)

[47]. Ces critères s'intéressent plus particulièrement à la comparaison des contenus des régions, dont le but est de rechercher la correspondance entre les pixels dans les segmentations à comparer. L'autre manière consiste à calculer la distance (euclidienne ou autre) entre les segments qui forment les frontières des régions. Nous citons ici le critère BDE (*Boundary Displacement Error*)[23]. Chacun de ces quatre critères sera brièvement détaillé à la section 7.5.

## 7.4 Vérité-terrain

Notre ensemble de base pour cette évaluation est la base d'images choisie par le groupe de Berkeley [42]. Cette base est dénommée *Berkeley Segmentation Dataset and Benchmark* (BSDB) contenant 300 images naturelles de tailles  $481 \times 321$  pixels, et est divisée en deux : 200 images pour ajuster les paramètres d'un algorithme de segmentation, et un ensemble de 100 images de test, pour évaluer sa performance. Ces images représentant des scènes naturelles avec au moins un objet discernable. Les images choisies abordent divers sujets naturels et industriels (être-humains, animaux, sous-marins, paysages, constructions,...). Chaque image est segmentée manuellement par plusieurs personnes (entre 4 et 7 étudiants de l'université de Berkeley) en répondant à la requête suivante : *Diviser chaque image en régions, où chaque région représente des choses ou des parties de choses discernables dans l'image. Il est important que toutes les régions aient à peu près la même importance. Le nombre de régions dans chaque image est laissé à votre choix. Un nombre entre 2 et 30 semble raisonnable dans la plupart des cas.* Pour le processus d'optimisation des méta paramètres de notre algorithme de segmentation, nous avons utilisé comme référence les segmentations humaines des 200 images de l'ensemble d'entraînement de la BSDB, les valeurs optimales des paramètres sur cette ensemble d'images ont été les suivants : nombre de classe (noté  $N_{cls}=6$ ),  $\beta_r=1.4$ ,  $\Gamma=30$ ,  $nbiter_{max} = 4$ ,  $N_r=75$ .



1

1

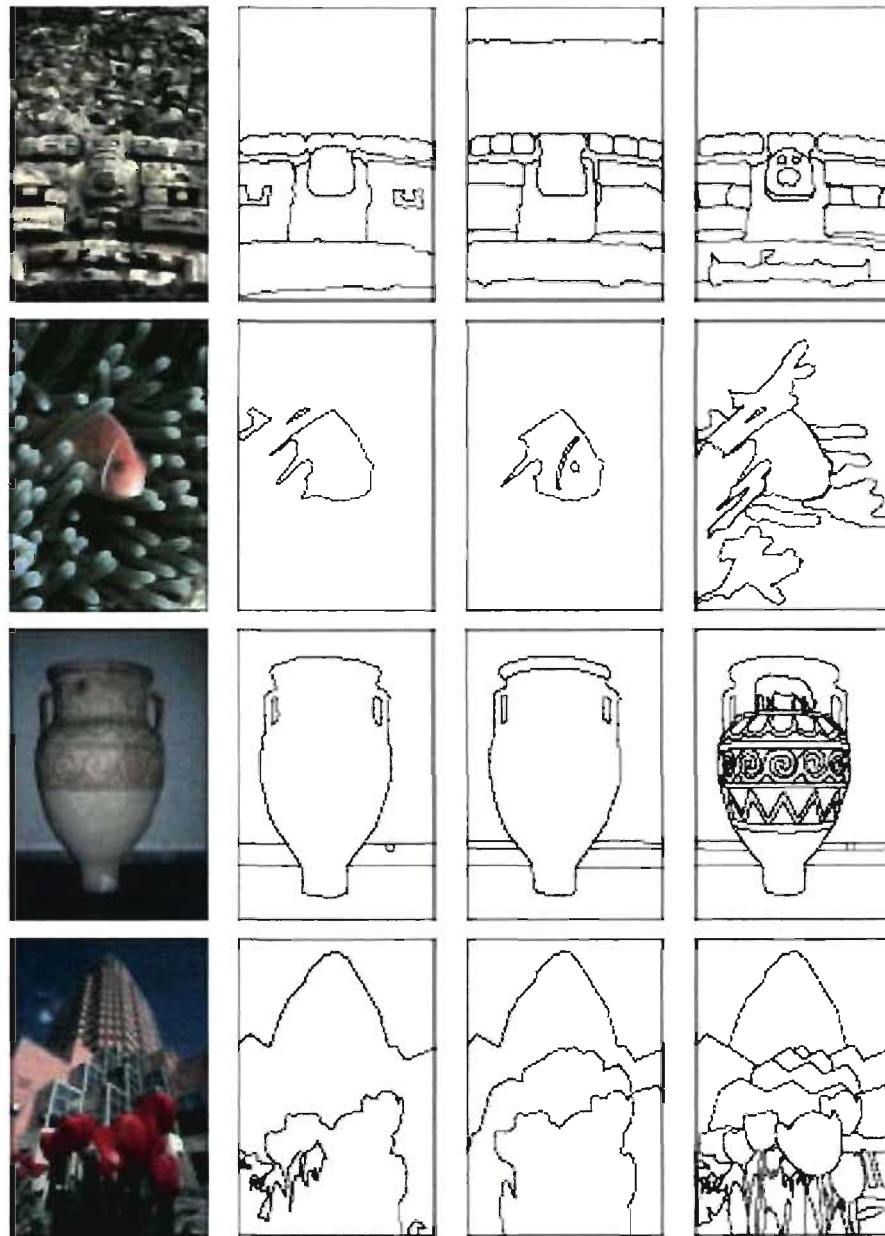


FIG. 7.2 – Segmentation humaine de quelques images de la BSDS

supervisés connus dans le milieu de l'imagerie, Mean-Shift [14], NCuts [62], et FII (*Felzenswald et Huttenlocker*) [22]. La comparaison est basée sur quatre mesures quan-



titatives de performance. L'une (BDE) est fondée sur l'appariement des frontières [23] et les autres sont fondées sur la différentiation entre régions dont nous proposons la mise en oeuvre par les critères suivants : VoI [47], GCE [42] et PRI [51]. Pour assurer une comparaison valide entre les différents algorithmes, nous avons suivi le même scénario établi par Yang [73]. Nous avons utilisé aussi toutes les 300 images (de la base que nous avons) normalisées afin que le plus grand côté soit égal en longueur ou en largeur à 320 pixels. Les paramètres de chacun des trois algorithmes avec lesquels nous avons évalué l'algorithme proposé dans ce mémoire, sont fixés à leurs valeurs optimales (voir [73]) et/ou correspondent aux valeurs internes proposées par l'auteur.

Dans les sections qui suivront, nous décrivons brièvement les différents critères utilisés pour l'évaluation.

## 7.6 Critères d'évaluation supervisée

### 7.6.1 PRI

Ce critère compte le nombre de paires de pixels ayant les mêmes étiquettes entre la segmentation machine et la segmentation de référence. Si on considère un ensemble  $K$  de segmentation humaine (vérité-terrain)  $\{S_1, S_2, \dots, S_k\}$  d'une même image  $X = \{x_1, x_2, \dots, x_N\}$ . Soit  $S_{test}$  la segmentation machine à comparer avec l'ensemble étiqueté manuellement (segmentations humaines). Nous notons par  $l_i^{S_{test}}$  l'étiquette de  $x_i$  dans la segmentation  $S_{test}$  et par  $l_i^{S_k}$  dans la segmentation  $S_k$ . Il est supposé que chaque étiquette  $l_i^{S_k}$  peut prendre des valeurs dans  $K_k$  (i.e., nombre de classes) et corrélativement  $S_{test}$  prend des valeurs dans  $L_{test}$  (étiquettes de la segmentation machine). La mesure de performance utilisée par *Caroline Pantofaru et al.* est le PRI. On juge qu'une segmentation est bonne si pour n'importe quelle paire de pixels  $x_i, x_j$  les étiquettes  $l_i^{S_{test}}, l_j^{S_{test}}$  de ces deux pixels, sont les mêmes (respectivement différentes) dans la segmentation machine et la segmentation humaine. On considère ainsi le *Probabilistic Rand (PR)* index :

$$PR(S_{test}, S_k) = \frac{1}{C_2^N} \sum_{i,j; i < j} [I(l_i^{S_{test}} = l_j^{S_{test}})p_{ij} + I(l_i^{S_{test}} \neq l_j^{S_{test}})(1 - p_{ij})] \quad (7.1)$$

Avec  $I(vrai) = 1$  et  $I(faux) = 0$ . Si on note que  $C_{ij} = I(l_i^{S_{test}} = l_j^{S_{test}})$  annonce le cas d'une paire de pixels  $i$  et  $j$  ayant la même étiquette dans la segmentation machine  $S_{test}$ . Le PR index peut aussi s'écrire

$$PR(S_{test}, S_k) = \frac{1}{C_2^N} \sum_{i,j; i < j} [C_{ij}p_{ij} + (1 - C_{ij})(1 - p_{ij})] \quad (7.2)$$

Avec  $N$  est le nombre de pixels, et  $P_{ij}$  est la probabilité de vérité-terrain pour que  $I(L_i = L_j)$ . En pratique  $P_{ij} = \frac{1}{K} \sum I(l_i^{S_k}, l_j^{S_k})$ , la relation moyenne d'une paire de pixels ayant la même étiquette parmi les segmentations humaines.

Cette mesure peut être facilement étendue à une autre mesure, le PRI (*Probabilistic Rand Index*) qui est en fait la moyenne des PR sur toutes les segmentations manuelles d'une image donnée [68].

### 7.6.2 Variation de l'information (VoI)

Contrairement au PRI qui est basé sur la relation entre les paires de pixels dans deux segmentations à comparer, le VoI [46][47] exprime la relation entre un point et son cluster. L'auteur définit la distance entre deux segmentations  $S_1$  et  $S_2$  comme étant la moyenne de l'entropie conditionnelle de l'une des deux segmentations étant donnée l'autre. Il utilise l'information mutuelle pour approximer la distance entre deux clusters à travers les différentes possibilités de clustering. Plus précisément, il mesure la quantité d'information qu'on peut gagner ou perdre si on passe d'un regroupement à un autre. Cette mesure est appelée variation de l'information VoI qui prend ses valeurs dans l'intervalle  $[0, \infty]$  (plus cette valeur est petite, plus cette mesure est bonne) :

$$VoI(S_1, S_2) = H(S_1) + H(S_2) - 2I(S_1, S_2) \quad (7.3)$$

Avec  $I(S_1, S_2) = \min\{H(S_1), H(S_2)\}$

et  $H(S_i)$  est l'entropie d'une segmentation  $S$  :

$$H(S) = - \sum_k P(k) \log P(k) \quad (7.4)$$

Tel que :  $P(k) = \frac{l_k}{l}$ , avec  $l$  est le nombre d'éléments d'une segmentation  $S$  et  $l_k$  le nombre d'éléments dans la région  $k$ . Si les deux segmentations  $S_1$  et  $S_2$  sont identiques, alors  $I(S_1, S_2) = H(S_1) = H(S_2)$ , ce qui veut dire que  $VoI(s_1, s_2)$  est nul. Pour plus de détails, voir [46][47].

### 7.6.3 Erreur de la cohérence globale (GCE)

L'auteur dans [42] mesure la cohérence entre deux segmentations humaines ou plus (cf. figure 7.2) d'une même image, et l'utilise par la suite pour comparer deux segmentations, l'une de référence ( $S_1$ ) et l'autre obtenue par une machine ( $S_2$ ). Cette mesure est basée sur deux erreurs en chaque pixel  $p$  et fournit une valeur  $e$  dans l'intervalle  $[0, 1]$  où 0 signifie « pas d'erreur ». Si un pixel  $p$  appartient à la région  $R(S_1, p)$  dans la segmentation  $S_1$ , et à la région  $R(S_2, p)$  dans la segmentation  $S_2$ , la première erreur de  $R(S_1, p)$  par rapport à  $R(S_2, p)$  est

$$E(S_1, S_2, p) = \frac{|R(S_1, p) \setminus R(S_2, p)|}{|R(S_1, p)|} \quad (7.5)$$

Et la deuxième erreur de  $R(S_2, p)$  par rapport à  $R(S_1, p)$  est

$$E(S_2, S_1, p) = \frac{|R(S_2, p) \setminus R(S_1, p)|}{|R(S_2, p)|} \quad (7.6)$$

Avec  $\setminus$  désigne la différence ensembliste, et  $|x|$  désigne le cardinal de l'ensemble  $x$ .  $E(S_1, S_2, p_i)$  tends vers 0 si  $R(S_1, p_i)$  est un sous-ensemble de  $R(S_2, p_i)$  et tends vers 1 si l'intersection des deux régions est réduite au pixel  $p_i$ .

Notons que cette mesure d'erreur locale est non symétrique. Elle encode la mesure de ressemblance dans une seule direction. Deux façons naturelles pour combiner les valeurs en une mesure d'erreur pour la totalité de l'image à fin de mesurer la dissimilarité entre segmentation machine et segmentation de référence.

– Erreur locale de cohérence :

$$LCE(S_1, S_2) = \frac{1}{n} \sum_i \min\{E(S_1, S_2, p_i), E(S_2, S_1, p_i)\} \quad (7.7)$$

– Erreur globale de cohérence :

$$GCE(S_1, S_2) = \frac{1}{n} \min\left\{\sum_i E(S_1, S_2, p_i), \sum_i E(S_2, S_1, p_i)\right\} \quad (7.8)$$

Comme  $LCE < GCE$  pour n'importe quelle deux segmentations, il est claire que  $GCE$  est plus sévère que  $LCE$ . Sur la figure 7.3,  $GCE$  permet une division simple (sous-segmentation), tandis que  $LCE$  permet une division mutuelle (mélange de sur et sous-segmentation) de l'image.

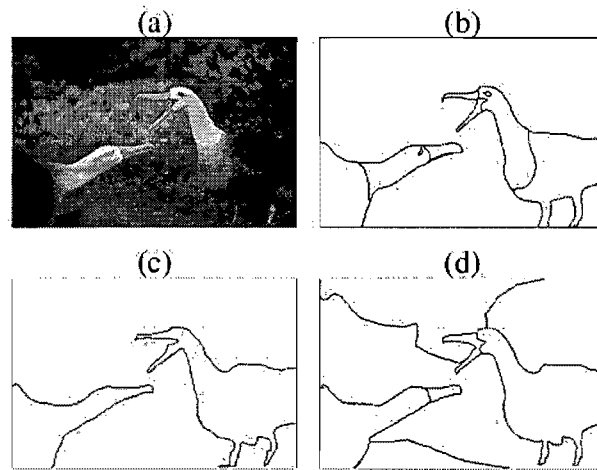


FIG. 7.3 – Tolérance de mesures d'erreur de segmentation : (a) image originale. (b)-(d) trois segmentations par différents humains. (b) et (d) deux segmentations illustrant la division mutuelle. (c) simple division par GCE

#### 7.6.4 Appariement des frontières (BDE) :

Plusieurs mesures effectuent la comparaison des frontières entre deux segmentations ou plusieurs, et calculent quelques éléments statistiques, comme par exemple l'erreur moyenne de déplacement des pixels des frontières d'une segmentation par rapport aux pixels des frontières de l'autre segmentation [23]. Dans [42], *Martin et al.* ont proposé une solution pour l'appariement des frontières de segmentation, par une approximation de graphe bipartite, en calculant le pourcentage d'appariement des arêtes de frontières en utilisant la moyenne harmonique de la *précision/rappel* [57][41], issu du domaine de l'indexation des données [58].

### 7.7 Vérification quantitative

Nous avons directement utilisé le programme (*code Matlab*) de comparaison réalisé par *Allen Y. Yang* [73]. Le tableau 7.1 montre les résultats de comparaison effectuée sur les 300 images de la BDSB entre les 4 algorithmes choisis. Le PRI varie entre 0 (qui signifie une mauvaise segmentation) et 1 (qui signifie une segmentation de parfaite qualité). VoI varie entre 0 et  $\infty$ , GCE entre 0 et 1, et BDE entre 0 et  $\infty$ , dans ces derniers cas une petite mesure signifie une segmentation de meilleur qualité.

Le tableau montre que l'algorithme proposé dans le cadre de ce mémoire, est beaucoup plus performant que le Mean-Shift, NCuts et FH, au sens de la moyenne du BDE sur l'ensemble des 300 images de la BDSB. Il est aussi meilleur que les deux premiers algorithmes et presque égale à FH au sens de la moyenne du PRI sur la même base d'images. On peut aussi interpréter le résultat en termes de la différence entre les quatre indices de segmentation. Les indices GCE et BDE permettent une division grossière de l'image et ne s'intéressent pas beaucoup aux détails (cf. figure 7.3(c)). Autrement-dit, ils s'intéressent plus particulièrement aux frontières extérieures des classes représentant les objets discernables de l'image à segmenter (le mauvais score de GCE par exemple est obtenu lorsque chaque pixel est assigné à une seule région).

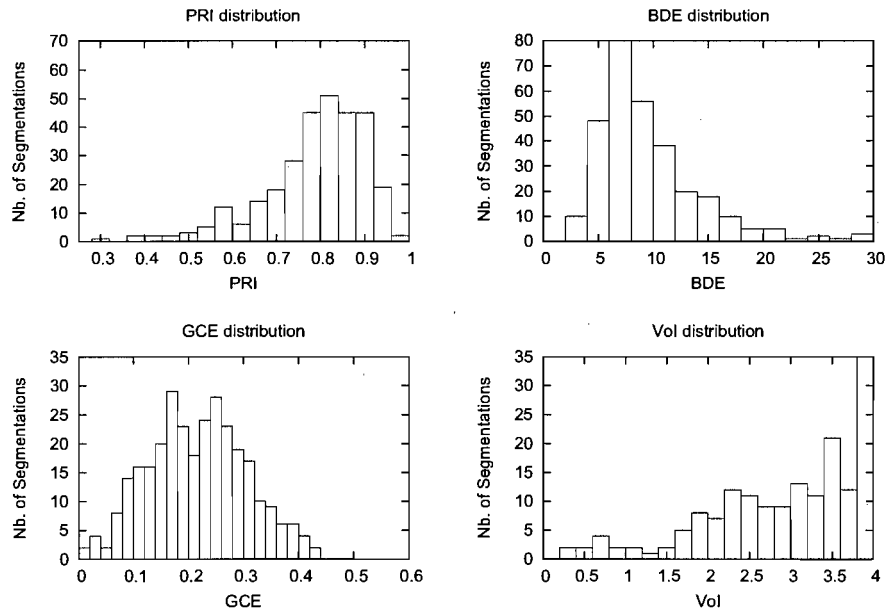


FIG. 7.4 – Distribution des différentes mesures de performance sur les 300 images de la base de Berkeley pour la configuration :  $N_{cls} = 6$  (*nombre de classes*),  $F=15 \times 15$ ,  $\check{F}=5 \times 5$ ,  $\check{G}=5 \times 5$ ,  $nbiter_{max} = 4$  avec  $\beta_r = 1.40$  et  $\tau_r = 25$

Méthodes	Indices de mesure			
	PRI	VoI	GCE	BDE
$HMC_{[1]}$	<b>0.7835</b>	3.9900	0.2900	<b>9.5700</b>
$HMC_{[2]}$	0.7816	3.8700	0.3000	<b>8.9300</b>
$HMC_{[3]}$	0.7714	3.6000	0.2900	9.0000
Humain	0.8754	1.1040	0.0797	4.9940
Mean-Shift	0.7550	2.4770	0.2598	9.7001
NCuts	0.7229	2.9329	0.2182	9.6038
FH	0.7841	2.6647	0.1895	9.9497

TAB. 7.1 – Mesures de performances des méthodes de segmentation.  $HMC_{[2]}$  est l'algorithme proposé avec les paramètres :  $N_{cls} = 6, \beta_r = 1.4, \tau_r = 25, \Gamma = 30, nbiter_{max} = 4, N_r = 75$ , pour  $HMC_{[1]}$ , nous avons utilisé les mêmes paramètres excepté  $N_r$ ,  $HMC_{[3]}$  utilise la méthode des vecteurs propres avec les paramètres :  $N_{cls} = 6, \beta_s = 1.0, \tau_s = 15, \Gamma = 30, nbiter_{max} = 4, N_r = 75$

D'après les expériences, les deux autres indices (PRI et VoI) semblent très corrélés avec la segmentation humaine (en termes de la perception visuelle humaine).

## 7.8 Influence des paramètres

Dans ce chapitre nous proposons d'étudier l'influence des paramètres ajustables sur les résultats de la méthode de segmentation proposée dans ce mémoire.

### 7.8.1 Nombre de classes

Choisir un nombre petit (ordre de 3 et moins) de classes (i.e., noté  $N_{cls}$ ) conduit à une sous-segmentation de certaines images, et choisir un nombre élevé (ordre de 10 et plus) conduit à un surplus de régions dans la segmentation finale. A cet effet, nous

Nombre de classes	Indices de mesure			
	PRI	VoI	GCE	BDE
5	0.7712	<b>3.4900</b>	0.2969	10.67
6	<b>0.7835</b>	3.8700	0.2900	9.5700
7	0.7808	3.7300	0.2900	<b>9.3600</b>
9	0.7790	4.0600	<b>0.2800</b>	10.230

TAB. 7.2 – *Effet du nombre de classes sur la segmentation*

avons fixé un nombre de classes, égale à 6 pour toutes les images de la BSD5. Nous avons constaté que ce nombre a participé en moyenne, à donner le meilleur score de l'évaluation des résultats (voir figure 7.5 et tableau 7.2).

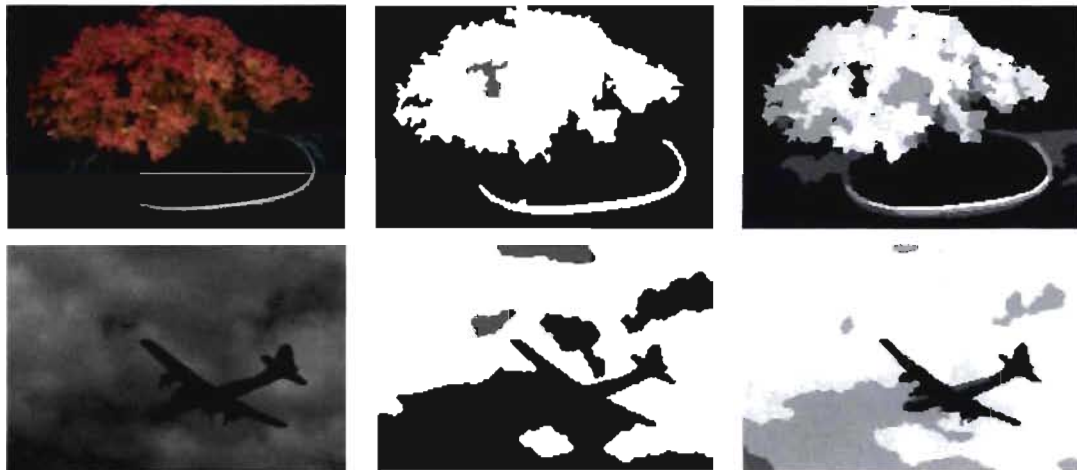


FIG. 7.5 – Influence du nombre de classes : à gauche 3 classes, à droite 7 classes

### 7.8.2 Paramètres d'inflation du MCL

Le paramètre d'inflation  $r$  utilisé par le MCL (chapitre 5) est très important. Ce paramètre a pour rôle d'élever les valeurs de la matrice de similarité de régions à la puissance de  $r$ , par elle même, afin d'amplifier les valeurs fortes et d'annuler les



valeurs faibles en ainsi de permettre aux régions de se détacher les une des autres. Ce processus mène à fusionner ensemble les régions reliées par des valeurs fortes (dans la matrice de similarité) et de séparer les régions reliées par des valeurs faibles. Pratiquement, la valeur initiale de ce paramètre est égale à 1.4 et la valeur finale doit être strictement supérieur à 1 .



FIG. 7.6 – Influence du paramètre d'expansion  $r$  : de gauche à droite, image originale,  $MCL_{r=1.2}$  (10 régions),  $MCL_{r=1.5}$  (17 régions)

### 7.8.3 Paramètre de segmentation par les vecteurs propres

Nous avons vu dans le chapitre 5, que les vecteurs propres de la matrice de similarité des régions sont généralement creux, c'est-à-dire que les premiers vecteurs propres (qui correspondent aux premières plus grandes valeurs propres), contiennent quelques composantes significatives (indiquant les régions à fusionner) et le reste des composantes sont quasi-nulles. Un seuil  $s$  est utilisé pour seuiller les composantes significatives des composantes nulles. Dans notre algorithme, nous multiplions simplement la plus grande composante de chaque vecteur propre, par une valeur réelle positive inférieure à 1. Le résultat de la multiplication sera considéré comme le seuil  $s$  de séparation. Dans le cadre des expériences effectuées sur les images de la BSDb, le seuil varie en moyenne entre 0.15 et 0.5.

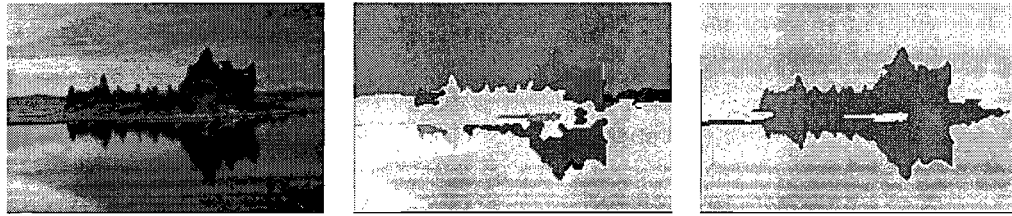


FIG. 7.7 – Influence du seuil  $s$  sur la séparation des composantes des vecteurs propres : de gauche à droite, image originale,  $EIGEN_{s=0.4}$  (28 régions),  $EIGEN_{s=0.9}$  (21 régions)

#### 7.8.4 Taille des régions à filtrer avant de lancer le partitionnement par graphe

Avant de procéder à la fusion des régions par partitionnement du graphe, on privilégie tout d'abord l'absorption des petites régions (i.e., celles dont la taille (noté  $\Gamma$ ) est inférieure à la taille du plus petit objet structurant que l'on désire discerner dans l'image) par la région adjacente avec laquelle elle est la plus connectée. Cette taille est considérée comme un paramètre à contrôler. L'opération de filtrage permet d'éliminer les petites régions non significatives et de diminuer ainsi la taille de la matrice d'affinité (i.e., gain dans son temps de traitement par l'algorithme de partitionnement par graphe).

#### 7.8.5 Temps de calcul

Notre algorithme de segmentation a été implanté sur une machine de type PC avec un double processeur Intel cadencé de 1 GHz et doté d'une mémoire centrale de taille 1Go. Nous avons constaté que la grande partie du temps de calcul est consommée durant la partie de construction du graphe d'adjacence. En moyenne, chaque image segmentée dans ce projet, dont la taille est de  $200 \times 133$  pixels, nécessite environs 1 minute et 20 secondes. Nous sommes devant un choix de compromis entre le temps

de calcul et la précision de la segmentation. Si nous voulons que la construction du graphe d'adjacence de régions soit plus rapide, nous préférons choisir des pas de balayage (décrit dans le chapitre 5), plus grand, mais la segmentation sera moins bonne. Le calcul de la distance entre deux régions, dépend fortement du nombre de pixels visités, autrement dit, pour que la distance entre deux régions, soit plus précise il faut que tous les pixels (pas=1) interviennent dans la comparaison des imagerie.

## Chapitre 8

### Discussion et conclusion

Dans ce mémoire, nous avons présenté une méthode originale de segmentation d'images couleur basée sur une *sur-segmentation* donnée par une segmentation Markovienne puis par une segmentation par partitionnement de graphe qui permet, d'une itération à l'autre, la fusion de certaines régions pour finalement représenter les différents objets discernables de l'image. Néanmoins, l'algorithme proposé a quelques limitations en particulier dans le cas de la segmentation d'images qui contiennent des animaux avec une texture de camouflage. Par exemple, sur la figure 8.1, il est difficile pour l'algorithme de segmenter le crocodile de l'arrière-plan parce que les deux plans avant (crocodile) et arrière ont quasiment deux textures similaires, ce qui rend (dans une certaine mesure) la perception du crocodile très délicate même par l'oeil. Du point de vue algorithmique, lors de la segmentation par la méthode Markovienne d'une image avec présence de camouflage, les régions appartenant au premier plan et les régions appartenant à l'arrière plan, auront quasiment des étiquettes similaires ce qui mène à la confusion entre certaines régions des différents objets.

D'autre part, puisque la segmentation Markovienne de cet algorithme s'opère premièrement sur la carte en niveaux de gris de l'image couleur, l'algorithme souffre d'un autre problème de confusion causé par le passage d'une image couleur à une carte en niveaux de gris. Ce passage est suivi dans la plupart des cas, par une perte d'informa-

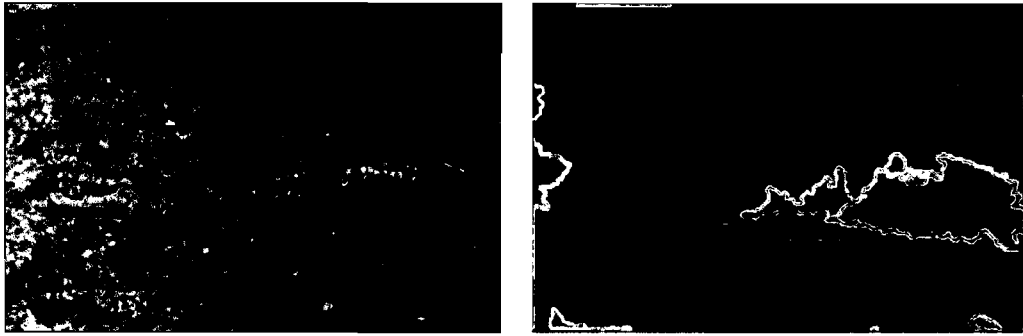


FIG. 8.1 – Effet de camouflage dans l'image couleur sur la segmentation

tion. Sur la figure (8.2), l'image en couleur (à gauche) est convertie tout d'abord en une carte de niveaux de gris (au milieu), puis la segmentation Markovienne de cette carte en 5 classes, produit une carte de régions libellées en couleurs (à droite), chaque étiquette représente une classe. Malheureusement, comme nous le remarquons sur la carte de milieu, les nuances de gris des pyramides et du ciel sont presque identiques impliquant la même étiquette pour les régions correspondantes à la partie inférieur du ciel et des pyramides. Ce phénomène contredit la segmentation de l'avant-plan (pyramides) de l'arrière-plan (ciel). Nous avons ainsi pensé à augmenter le nombre de classes lors de la segmentation Markovienne pour donner la chance à ces deux plans d'avoir des étiquettes différentes, mais malheureusement, ce n'est pas toujours le cas. En plus si nous choisissons un nombre élevé de classes, nous nous trouverons devant un autre problème qui est le surplus de régions. Ce problème est la source principale du gaspillage du temps durant la phase de construction du graphe d'adjacence de régions. En plus, la matrice de similarité de régions devient importante et sa manipulation par les vecteurs propres et/ou le MCL devient très exigeante en matière de temps de calcul.

Nous avons aussi constaté que le résultat de la méthode de segmentation proposée dans le cadre de ce projet dépend très fortement de la segmentation Markovienne de départ, autrement-dit, si nous arrivons à diviser les différentes classes de l'image



FIG. 8.2 – Problème de confusion des objets de l'image lors du passage de la base couleur en carte de niveaux de gris

en régions bien séparées (pas de mélange entre les régions des différents objets de l'image), nous pouvons par la suite appliquer l'algorithme de regroupement de ces régions (MCL ou EIGEN) sans crainte (voir figure 8.3).

Comme perspective, afin d'améliorer la qualité de la méthode de la segmentation proposée, nous avons pensé à plusieurs stratégies. Premièrement, nous pensons que la segmentation Markovienne appliquée directement sur l'image couleur permettrait d'améliorer la segmentation finale, parce que les régions dans ce cas de segmentation ne sont pas relativement mélangées au départ, ce qui permet au programme (de fusion des régions) de fonctionner sur une base de régions appartenant à des objets (classes) différentes. Deuxièmement, nous pensons aussi que l'application de la segmentation sur une autre base de chrominance par exemple la saturation permettrait de distinguer entre des régions de différentes classes (voir figure 8.4) permettant une plus grande facilité au niveau du regroupement des régions. Troisièmement, le réglage minutieux des paramètres joue aussi un rôle important dans la qualité de segmentation. Jusqu'à maintenant nous n'avons pas vraiment trouvé les paramètres avec lesquels la qualité de segmentation augmente au-delà de ce que nous avons présenté dans l'évaluation. La *re-segmentation* basée sur le regroupement des régions par des méthodes de partitionnement de graphes est prometteuse car celle-ci agit hiérarchiquement comme le fait le système visuel humain.

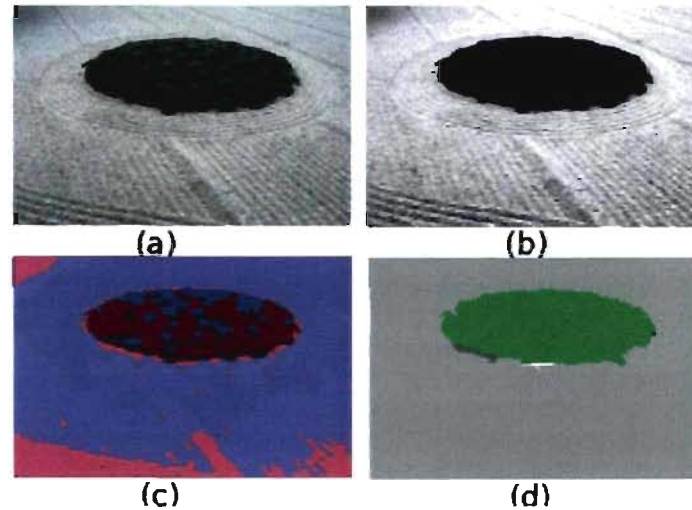


FIG. 8.3 – Effet de la segmentation Markovienne sur le processus de regroupement de régions et la segmentation finale : (a) image originale, (b) image en niveaux de gris, (c) carte de segmentation Markovienne (165 régions), (d) carte de segmentation finale par regroupement de régions  $MCL_{N_r=4}$  (4 régions)



FIG. 8.4 – Segmentation sur la base chromatique de saturation : à gauche, segmentation ICM de l'image 8.2, à droite, regroupement par MCL.

# Bibliographie

- [1] M.N. Ahmed, S.M. Yamany, N.A. Mohamed, and A.A. Farag. A modified fuzzy c-means algorithm for MRI bias estimation and adaptative segmentation. *Proc. Second International Conference on Medical Image Computing and Computer Assisted Intervention, MICCAI'99 Cambridge (England)*, 19(22) :72–81, sept 1999.
- [2] L. Baum, T. Petrie, G. Soules, and N. Weiss. A maximisation technique occurring in the statistical analysis of probabilistic function of Markov chains. *Annals of Mathematical and Statistics*, 41(1) :164–171, 1970.
- [3] B.J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of Royal Statistic Society*, 26(2) :192–236, 1974.
- [4] J. Besag. Statistical analysis of dirty picture. *Journal of Royal Statistic Society*, 48 :259–302, 1986.
- [5] M. Borsotti, P. Campadelli, and Schettini. Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*, 19 :741–747, 1998.
- [6] P.S. Bradeley and Usama M. Fayed. Refining initial points for k-means clustering. *Proceeding of the Fifteenth International Conference on Machine Learning*, pages 91–99, Juillet 1998.
- [7] J.S. Cardoso and L. Corte-Real. Toward a generic evaluation of image segmentation. *IEEE Transactions on Image Processing*, 14(11) :1773–1782, 2005.
- [8] G. Celeux and D. Diebolt. The SEM algorithm : a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational*



- Statistics Quarterly*, 2(1) :73–82, 1985.
- [9] G. Celeux and G. Govaert. A classification EM algorithm for clustering and two stochastic versions. *Rapport de Recherche 1364 Inria*, 1991.
- [10] S. Charber, C. Resenberger, and B. Emile. Évaluation de la performance de la segmentation d’images par fusion de critères, Labo. vision et robotique-UPRES EA 2078 ENSI de bourges, Univ. d’Orléans.
- [11] S. Charbier, C. Rosenberger, H. Laurent, B. Emil, and P. March. Evaluating the segmentation result of a gray-level image. *12th European Signal Processing Conference (EUSIPCO’04)*, Vienna, Austria, pages 953–956, September 2004.
- [12] J.P. Cocquerez. *Analyse d’images : filtrage et segmentation*. S. Philip editors, Masson, Paris, 1995.
- [13] P. Colantoni. Contribution des structures de données à la segmentation d’images couleur. Élaboration d’un outil d’infographie textile. *Thèse de doctorat, Université Saint-Etienne, France*, 1998.
- [14] D. Comaniciu and P. Meer. Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 :603–619, 2002.
- [15] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1) :1–38, 1977.
- [16] R. Deriche. Using canny’s criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2) :167–187, 1987.
- [17] E. Diday, J. Lemaine, J. Pouget, and F. Testu. Éléments d’analyse de données, Paris. (85), 1982.
- [18] F. Didos and G. Koepfler. Segmentation d’images couleur par méthodes variationnelles. *16ième Colloque GRETSI, Grenoble, France*, pages 367–370, 1997.
- [19] S. Van Dongen. Graph clustering by flow simulation. *Thèse doctorat, Université d’Utrecht, Netherlands*, Mai 2000.

- [20] P.A. Arbelaez Escalante. Une approche métrique pour la segmentation d'images. *Thèse de doctorat, Univ. Paris Dauphine U.F.R. Mathématique de la décision. France*, 2005.
- [21] A. Tremeau et al. *Image numérique couleur*. Dunod, 2004.
- [22] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision.*, 59 :167–181, 2004.
- [23] J. Freixenet, X. Munos, D. Rabat, J. Martin, and X. Cuff. Yet another survey on image segmentation. *7th European Conference on Computer Vision*, (3) :408–422, 2002.
- [24] J. Diebolt G, Celeux. L'algorithme SEM : un algorithme d'apprentissage probabiliste pour la reconnaissance de mélange de densités. *Revue de Statistiques appliquées*, 34(2) :35–52, 1986.
- [25] I. Gath and A.B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7) :773–781, 1989.
- [26] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesienne restoration of images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 6 :721–741, 1984.
- [27] S. Geman and C. Graffigne. Markov random field models and their application to computer vision. *Proceedings of the International Congress of Mathematicians, Berkeley California, USA*, pages 1496–1517, Nov 1986.
- [28] R.M. Haralik. Digital step edges from zero-crossing of second directional derivative. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1) :58–68, 1984.
- [29] I.L. Herlin, D. Beriziat, G. Giraudon, C. Nguyen, and C. Graphigne. Segmentation of echocardiographic images with Markov random field. *Rocquencourt (France) INRIA Research report 2424*, page 18, 1994.

- [30] A. Hernandez. Fusion d'images radiographiques et ultrasonores, visualisation stéréoscopique : application aux pathologie du sein. *Thèse de doctorat, Institut National des sciences appliquées de Lyon*, 1997.
- [31] M.F. Hueckel. An operator which locates edges in digitized pictures. *Journal Association on Computer Machine*, 18(1) :113–125, 1971.
- [32] H.J. Huisman and J.M. Thijssen. Adaptative texture feature extraction with application to ultrasonic image analysis. *Ultrasonic image analysis*, 20 :132–148, 1998.
- [33] Z. Kato. Modélisation markovienne multiresolution en vision par ordinateur, application à la segmentation d'images SPOT. *Thèse de Doctorat, Univ. de Nice Sophia Antipolis*, 1994.
- [34] S. Kirkpatrick, C.D. Gellat, and M.P. Vecchi. Optimisation by simulated annealing. *Science*, 220(4598) :671–680, 1983.
- [35] R. Kothari and D. Pitts. On finding the number of clustering. *Pattern Recognition Letters*, 20(20) :405–416, 1999.
- [36] M. Levine and A. Nazif. Dynamic measurment of computer generated image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(25) :155–164, 2005.
- [37] J. Liu and Y.H. Yang. Multiresolution color image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7) :689–700, 1994.
- [38] W.Y. Liu, I.E. Magnin, and G. Gemen. Un nouvel opérateur pour la détection de ruptures dans des signaux bruités. *Traitement du signal*, 12(3) :225–236, 1995.
- [39] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistic and Probability, Berkeley, Univ. Of California Press*, 1 :281–297, 1967.
- [40] D. Marr and H. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London B207*, pages 187–217, 1980.

- [41] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5) :530–549, 2004.
- [42] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural image and its application to evaluating segmentation algorithms and measuring ecological statistic, Vancouver, Canada. *In Proceedings International Conference on Computer Vision (ICCV)*, 2001.
- [43] C. Mathieu. Segmentation d’images par pyramides souples : application à l’imagerie médicale multidimensionnelle. *Thèse de Doctorat, Institut national des sciences appliquées de Lyon-France*, 1993.
- [44] B. McCane. On the evaluation of image segmentation algorithms. *Digital Image Computing, Techniques and Application DICTA’97 and IVCN’Z97*, pages 455–459, October 1997.
- [45] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis : a survey. *Medical ImageAnalysis*, 1(2) :91–108, 1996.
- [46] M. Meila. Comparing clustering by the variation of information. *Conference on Learning Theory*, pages 173–187, 2003.
- [47] M. Meila. Comparing clustering : an axiomatic view. *In the 22nd Proceeding International Conference on Machine Learning*, 14, 2005.
- [48] N. Metropolis, A.W.Rosenbluth, M.W. Rosenbluth, and A.H. Teller. Equation of state calculation by fast computing machines. *The Journal of Chimical and Physics*, 21(6) :1087–1092, 1953.
- [49] F. Heitz P. Perez. Restriction of a markov random field on a graph and multi-resolution image analysis, Rennes (France) : Irisa. *Publication internal repport 784*, page 49, January 1994.
- [50] P. Palisson. Exploitation de paramètres de texture à l’aide de modèles connexionnistes. vers une extension de la notion de texture grace à l’oculométrie. *Thèse de doctorat, Institut national des sciences appliquées de Lyon*, 1994.

- [51] C. Pantofaru and M. Hebert. A comparison of image segmentation algorithms. *Technical Report CMU-RI-TR-05-40 CMU.*, 2(14), 2005.
- [52] P. Perez. Champs Markoviens et analyse multiresolution de l'image : application a l'analyse multiresolution, Rennes (France) : IRISA. *Publication Interne 713*, page 63, 1993.
- [53] S. Philip-Folguet and L. Guigues. Évaluation de la segmentation d'images : état de l'art, nouveaux indices et comparaison. *Technical report, ETIS.*, mars 2001.
- [54] O. Pichler, A. Teuner, and B. Hosticka. An unsupervised texture segmentation algorithm with feature space reduction and knowledge feedback. *IEEE Transaction on Image Processing*, 7(1) :53–61, 1998.
- [55] W. Pieczynski. Statistical image segmentation. *Machine Graphics and Vision*, 1(1/2) :261–286, 1992.
- [56] C.A. Poynton. Frequently asked Question about Gamma. <http://www.inforamp.net/poynton>.
- [57] I.E. Abdou W.K. Pratt. Qualitative design and evaluation of enhancement/thresholding edge detector. *IEEE Transaction*, 67(5) :753–763, Mai 1979.
- [58] V. Van Rijsberger. Information retrieval. *Department of Computer Science Univ. of Glasgow*, 1979.
- [59] M.C. Robini. Inversion Bayesienne et algorithmes stochastiques de type recuit, application à la restauration d'images et la la reconstruction 3D à partir d'un nombre très limité de projections. *Thèse de Doctorat, Institut National des Science appliquées de Lyon-France*, 1999.
- [60] C. Rouquet, P. Bonton, and R. Tomezak. étude comparative des stratégies de segmentation non supervisée en régions par champs de Markov. *Traitement du signal*, 15(1) :39–55, 1998.
- [61] D. Rueckert, P. Burger, M. Forbat, R.D. Mohiaddin, and G.Z. Yang. Automatic tracking of the aorta in cardiovascular MR images using deformable models. *IEEE Transaction on Medical Imaging*, 16(5) :581–590, 1997.

- [62] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico*, pages 731–737, Juin 1997.
- [63] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. *IEEE International Conference on Computer Vision, India*, pages 154–160, Janvier 1998.
- [64] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8), Aout 2000.
- [65] C. Soubiran, G. Celeux, J. Diebolt, and C. Robert. Analyse de mélange gaussien pour de petits échantillons : application à la cinématique stellaire. *Revue de Statistique appliquée*, 39(3) :17–35, 1991.
- [66] X. Y. Stella and J. Shi. Segmentation given partial grouping constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2), Février 2004.
- [67] K.C. Strasters and J.J. Gerbrands. Three-dimentional image segmentation using a split, merge and group approche. *Pattern Recognition Letters*, 12 :307–325, 1991.
- [68] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 29 :929–944, Juin 2007.
- [69] V. Vapnik. The nature of statistical learning theory. *Springer-Verlag New York, Inc.*, 1995.
- [70] L. Vinet. Segmentation et mise en correspondance de régions de paires d’images stéréoscopiques. *Thèse de Doctorat, Univ. de Paris IX Dauphine*, 1991.
- [71] Y. Wu. Segmentation : Clustering, Graph Cut and EM. *EECS 423-Advanced Computer Vision Notes Series 5*.
- [72] G. Wyszecki and W. Stiles. *Color Science : Concepts and Methods, Quantitative Data Formulate*. John Wiley and Sons, New York, Second edition, 1982.

- [73] A.Y. Yang, J. Wright, and S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, janvier 2007.
- [74] J.F. Yao. Analyse d'images par champs de markov. *Institut de Recherche mathématique de Rennes, Univ. de Rennes 1* : [http ://www.univ.rennes1.fr/ yao](http://www.univ.rennes1.fr/~yao).
- [75] Y.J. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8) :1335–1346, 1996.

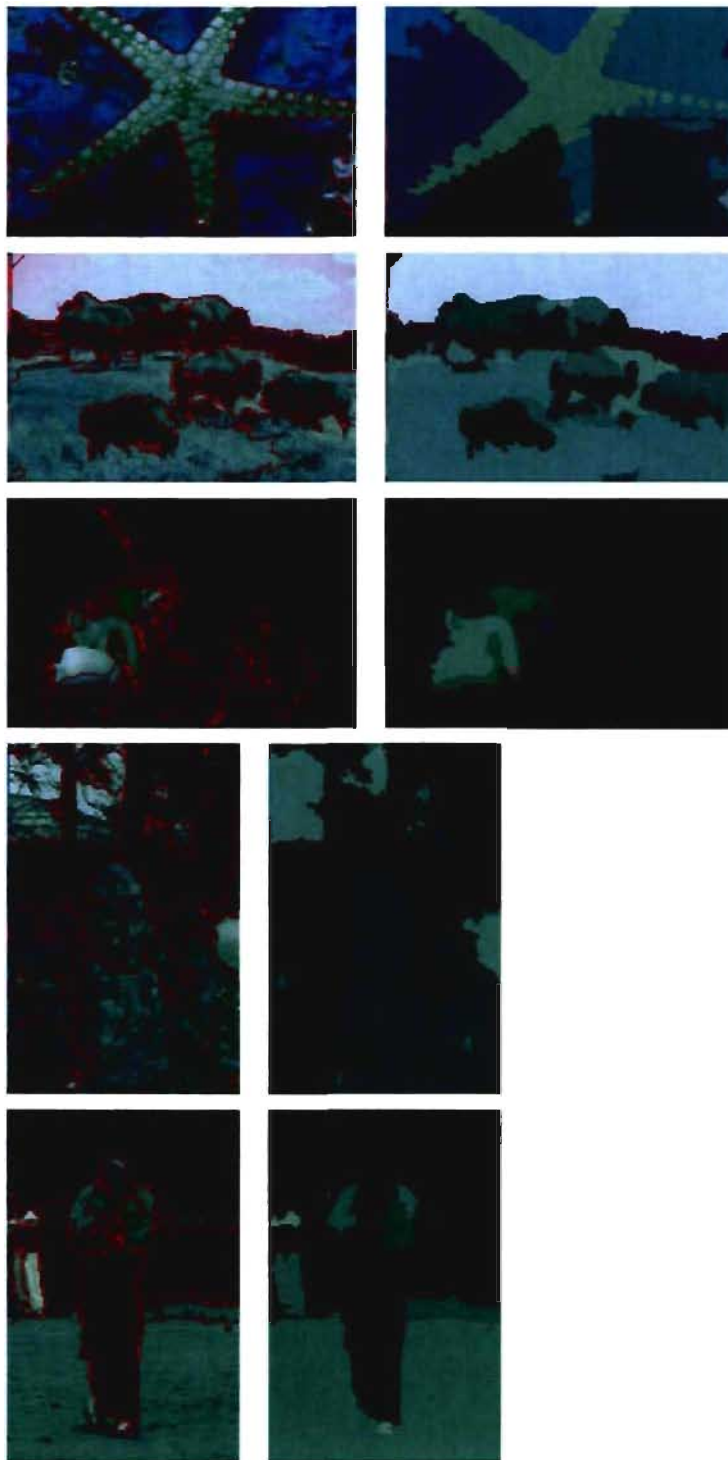
# Annexe A

## Résultats 1

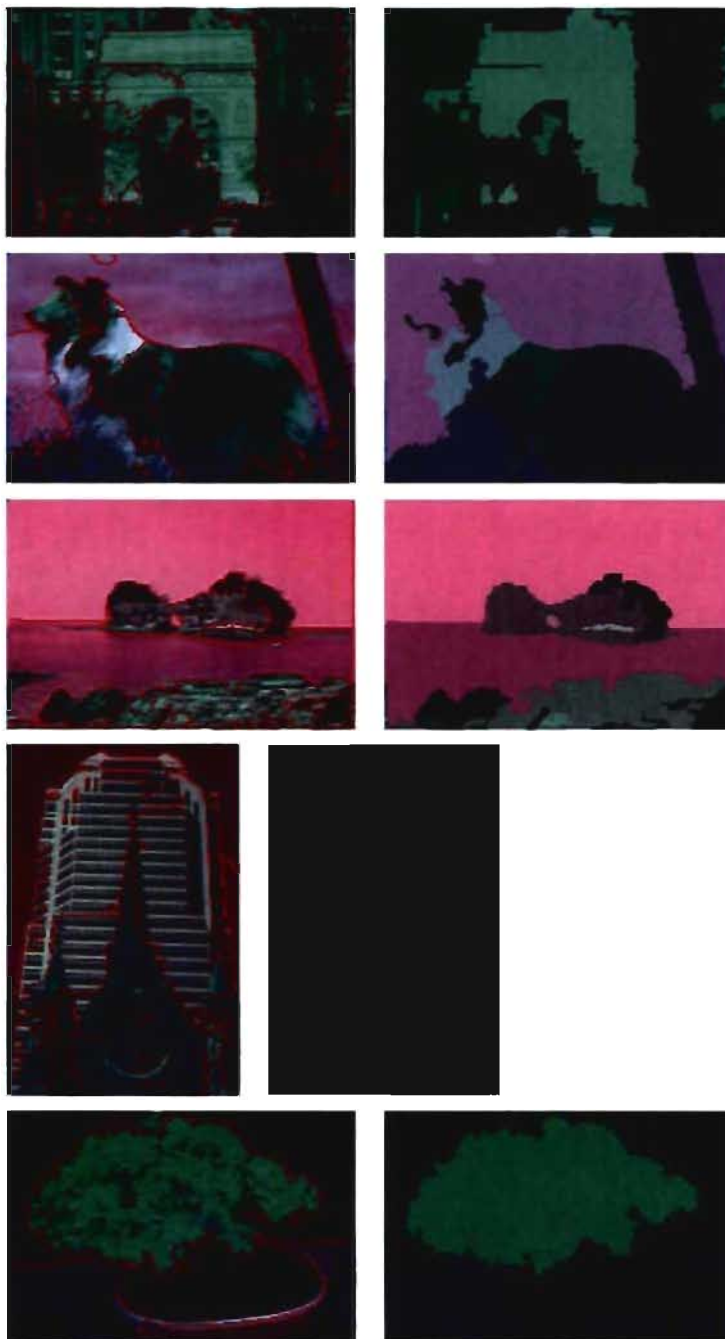
Les segmentations suivantes sont obtenues par l'algorithme proposé (*Partitionnement Markovien Hiérarchique* ou en anglais *HMC* pour *Hierarchical Markovian Clustering*) dont les paramètres sont :

1. Nombre de classes  $N_{cls} = 6$ .
2. Nombre d'itérations  $nbiter_{max} = 2$ .
3. sans *a priori* sur le nombre de régions dans la segmentation finale.
4. Le paramètre  $r$  est constant (1.5) dans toutes les itérations.











# Annexe B

## Résultats 2

Les segmentations suivantes sont obtenues par l'algorithme proposé (*Partitionnement Markovien Hiérarchique ou en anglais HMC pou Hierarchical Markovian Clustering*) dont les paramètres sont :

1. Nombre de classes  $N_{cls} = 6$ .
2. Nombre d'itérations  $nbiter_{max} = 4$ .
3.  $\beta_r = 1.4$ .
4.  $\tau = 25$ .
5. sans *a priori* sur le nombre de régions dans la segmentation finale.

